

# The automultiplechoice package\*

Alexis Bienvenue  
paamc@passoire.fr

December 10, 2021

## Abstract

This package helps designing multiple choice exams ready for automated marking from papers scans.

Answers and questions are optionally shuffled, creating different sheets for every student.

## 1 Introduction

The package `automultiplechoice` helps formatting multiple choice questionnaires with automated marking from papers scans in mind:

- The package can produce different copies of the question sheet for each student, optionally shuffling answers and questions for each student.
- Markers can be printed on each sheet, so as to be able to analyse scans after examination. All the needed information about the position of the markers and the boxes to be checked by the students is given in an auxiliary file during  $\text{\LaTeX}$  run.

See Auto Multiple Choice (AMC) software (<https://www.auto-multiple-choice.net/>) for an integration of this package, with user interface for automated marking.

## 2 Samples

We begin with several samples to see what can be done with the `automultiplechoice` package. All `automultiplechoice` commands and options will be detailed further.

For all these samples, two sets of questions are used: a group of geography questions, and a group of history questions. These are defined in a common  $\text{\LaTeX}$  file named `questions.tex`:

```
\element{geography}{  
  \begin{question}{Ghana}  
    What is the capital of Ghana?  
    \begin{choiceshoriz}  
      \correctchoice{Accra}  
      \wrongchoice{Addis Abeba}  
      \wrongchoice{Ankara}  
      \wrongchoice{Apia}
```

---

\*This document corresponds to version revision: `r:a9251468` from `AMC 1.5.2+git2022-10-23`

```

        \end{choiceshoriz}
    \end{question}
}

\element{geography}{
    \begin{question}{Thailand}
        What is the capital of Thailand?
        \begin{choiceshoriz}
            \correctchoice{Bangkok}
            \wrongchoice{Banjul}
            \wrongchoice{Beijing}
            \wrongchoice{Beirut}
            \wrongchoice{Berlin}
        \end{choiceshoriz}
    \end{question}
}

\element{geography}{
    \begin{question}{Egypt}
        What is the capital of Egypt?
        \begin{choices}
            \correctchoice{Cairo}
            \wrongchoice{Caracas}
            \wrongchoice{Cayenne}
            \wrongchoice{Chisinau}
            \wrongchoice{Conakry}
        \end{choices}
    \end{question}
}

\element{geography}{
    \begin{question}{Ireland}
        What is the capital of Ireland?
        \begin{multicols}{3}
            \begin{choices}
                \correctchoice{Dublin}
                \wrongchoice{Dili}
                \wrongchoice{Djibouti}
                \wrongchoice{Doha}
                \wrongchoice{Dakar}
                \wrongchoice{Dhaka}
            \end{choices}
        \end{multicols}
    \end{question}
}

\element{history}{
    \begin{questionmult}{1901}
        Which of the following events are taking place during the year

```

```

1901?
\begin{choices}
  \correctchoice{Funeral of Queen Victoria in London}
  \correctchoice{Official end of the Caste War of Yucat\'an}
  \wrongchoice{King George of Greece becomes absolute monarch of Crete}
  \wrongchoice{The first line of the Paris M\'etro is opened}
\end{choices}
\end{questionmult}
}

\element{history}{
  \begin{questionmult}{1850}
    Which of the following events are taking place during the year
    1850?
    \begin{choices}
      \correctchoice{American Express is founded by Henry Wells \& William Fargo}
      \wrongchoice{Napoleon Bonaparte crosses the Alps and invades Italy}
      \wrongchoice{Kwang-su becomes emperor of China}
      \wrongchoice{First horse-drawn omnibuses established in London}
    \end{choices}
  \end{questionmult}
}

\element{history}{
  \begin{questionmult}{1971}
    Which of the following events are taking place during the year
    1971?
    \begin{choices}
      \correctchoice{Apollo 14 lands on the Moon}
      \correctchoice{The Soviet Union launches Salyut 1}
      \correctchoice{Death of Louis Armstrong}
      \wrongchoice{The first commercial Concorde flight takes off}
    \end{choices}
  \end{questionmult}
}

```

We will ask automultiplechoice package to include two geography questions and two history questions at random for each student, shuffling questions and answers, with the following code:

```

\cleargroup{all}
\shufflegroup{geography}
\copygroup[2]{geography}{all}
\shufflegroup{history}
\copygroup[2]{history}{all}
\shufflegroup{all}
\insertgroup{all}

```

You can read these commands as “clear group **all**, shuffle questions inside group **geography** and copy the first two to group **all**, do the same for group **history**, shuffle the four questions copied into **all** and print them”.

## 2.1 Standard layout

A set of 30 students sheets can be produced from the following L<sup>A</sup>T<sub>E</sub>X source named `sample-amc.tex`:

```
\documentclass{article}
\usepackage{automultiplechoice}
\usepackage{multicol}
\begin{document}

\input{questions.tex}

\onecopy{30}{

\noindent{\bf AMC \hfill SAMPLE TEST}

\vspace{3ex}
```

For this test, package `{\sf automultiplechoice}` is used without any option. Page markers are printed in view of an automated marking from papers scans. DRAFT indications can be cancelled using `{\tt nowatermark}` option.

Commands from `{\sf automultiplechoice}` are used to print, for each student, two geography questions and two history questions, at random. Questions and answers are shuffled.

```
\vspace{3ex}

\cleargroup{all}

\shufflegroup{geography}
\copygroup[2]{geography}{all}
\shufflegroup{history}
\copygroup[2]{history}{all}
\shufflegroup{all}
\insertgroup{all}

}
\end{document}
```

producing a 30-pages document (every page has number 1), from which we show the first pages on page 7.

Note that “DRAFT” indications can be cancelled using option `nowatermark`, or using AMC software.

You can see on each page markers that can be used for automated completed answer sheets scans analysis:

- Four circles ● are printed in the corners, to be able to analyse any rotation or scaling of the scans.
- Binary boxes are printed in the header area, so as to be able to read student sheet number and page number. On page 2 for example, you can see that these binary boxes are coding 2/1/59:



+2/1/59+

Here, 2 is the student sheet number, 1 is the page number for this student, and 59 is a checking value that can be used for checking correct identification from a scan.

If you also use `calibration` option, `automultiplechoice` will produce a `.xy` file with informations about the exact position in the page of all the markers, and all the boxes. This option is automatically set by AMC software, which then use the information in the `.xy` file for automated marking.

## 2.2 Separate answer sheet

In some situations, you may need a separate answer sheet:

- this makes cheating even more difficult;
- this can reduce the number of pages to scan.

This is done using `separateanswersheet` option of `automultiplechoice` package. You also have to use commands `\AMCformBegin` to indicate the beginning of this separate answer sheet (usually after a `\clearpage` or `\AMCcleardoublepage` command), and `\AMCform` to insert the form to be completed by the students, as in the following example (`sample-separate.tex`):

```
\documentclass{article}
\usepackage[separateanswersheet]{automultiplechoice}
\usepackage{multicol}
\begin{document}
```

```
\input{questions.tex}
```

```
\onecopy{30}{
```

```
\noindent{\bf AMC \hfill SAMPLE TEST}
```

```
\vspace{3ex}
```

For this test, package `{\sf automultiplechoice}` is used with `{\tt separateanswersheet}` option, so that all answers are to be filled on a separate sheet by students. Page markers are printed in view of an automated marking from papers scans. DRAFT indications can be cancelled using `{\tt nowatermark}` option.

Commands from `{\sf automultiplechoice}` are used to print, for each student, two geography questions and two history questions, at random. Questions and answers are shuffled.

```
\vspace{3ex}
```

```
\cleargroup{all}
```

```

\shufflegroup{geography}
\copygroup[2]{geography}{all}
\shufflegroup{history}
\copygroup[2]{history}{all}
\shufflegroup{all}
\insertgroup{all}

```

```

\clearpage

```

```

\AMCformBegin

```

This is the answer sheet: all answers are to be ticked on this page to be taken into account.

```

\vspace{2ex}

```

```

\AMCform

```

```

}
\end{document}

```

First pages of the result are shown on page 8. There are now 2 pages per student: the first with questions, and the second for answers. Only the second will be completed by the students, and scanned for analysis.

## 2.3 Without markers

With the `nopage` option, package `automultiplechoice` does not include any page markers for scan processing. I'm afraid you can't use any automated marking software with this layout, but you can still use answer sheet and corrected answer sheet (option `indivanswers`, added here) for a manual marking...

The  $\text{\LaTeX}$  source `sample-plain.tex` that only differs from `sample-amc.tex` by its options passed to `automultiplechoice`:

```

\usepackage[nopage,indivanswers]{automultiplechoice}

```

produces a 30-pages document, from which we show the first pages on page 9.

# First pages from L<sup>A</sup>T<sub>E</sub>X source detailed in section 2.1 – see sample-amc.pdf

●  ● +1/1/60+

AMC SAMPLE TEST

For this test, package `automultiplechoice` is used without any option. Page markers are printed in view of an automated marking from papers scans. DRAFT indications can be cancelled using `\nocancelmark` option.

Commands from `automultiplechoice` are used to print, for each student, two geography questions and two history questions, at random. Questions and answers are shuffled.

**Question 1** Which of the following events are taking place during the year 1971?

☐ The Soviet Union launches Salyut 1  
☐ The first commercial Concorde flight takes off  
☐ Death of Louis Armstrong  
☐ Apollo 14 lands on the Moon

**Question 2** What is the capital of Egypt?

☐ Cayenne  
☐ Caracas  
☐ Cuito  
☐ Conakry  
☐ Chisinau

**Question 3** Which of the following events are taking place during the year 1850?

☐ Napoleon Bonaparte crosses the Alps and invades Italy  
☐ First horse-drawn omnibuses established in London  
☐ American Express is founded by Henry Wells & William Fargo  
☐ Kwang-on becomes emperor of China

**Question 4** What is the capital of Ghana?

☐ Accra ☐ Addis Ababa ☐ Ankara ☐ Apia

● For your examination, preferably print documents compiled from `automultiple-choice`. ●

●  ● +2/1/60+

AMC SAMPLE TEST

For this test, package `automultiplechoice` is used without any option. Page markers are printed in view of an automated marking from papers scans. DRAFT indications can be cancelled using `\nocancelmark` option.

Commands from `automultiplechoice` are used to print, for each student, two geography questions and two history questions, at random. Questions and answers are shuffled.

**Question 1** Which of the following events are taking place during the year 1901?

☐ The first line of the Paris Metro is opened  
☐ Official end of the Caste War of Yucatán  
☐ King George of Greece becomes absolute monarch of Creta  
☐ Funeral of Queen Victoria in London

**Question 2** What is the capital of Ireland?

☐ Djibouti ☐ Dhaka ☐ Dhaka  
☐ Dili ☐ Dili ☐ Dakar

**Question 3** What is the capital of Ghana?

☐ Apia ☐ Accra ☐ Addis Ababa ☐ Ankara

**Question 4** Which of the following events are taking place during the year 1850?

☐ Napoleon Bonaparte crosses the Alps and invades Italy  
☐ First horse-drawn omnibuses established in London  
☐ American Express is founded by Henry Wells & William Fargo  
☐ Kwang-on becomes emperor of China

● For your examination, preferably print documents compiled from `automultiple-choice`. ●

●  ● +3/1/52+

AMC SAMPLE TEST

For this test, package `automultiplechoice` is used without any option. Page markers are printed in view of an automated marking from papers scans. DRAFT indications can be cancelled using `\nocancelmark` option.

Commands from `automultiplechoice` are used to print, for each student, two geography questions and two history questions, at random. Questions and answers are shuffled.

**Question 1** Which of the following events are taking place during the year 1971?

☐ The Soviet Union launches Salyut 1  
☐ The first commercial Concorde flight takes off  
☐ Apollo 14 lands on the Moon  
☐ The Soviet Union launches Salyut 1  
☐ Death of Louis Armstrong

**Question 2** Which of the following events are taking place during the year 1850?

☐ First horse-drawn omnibuses established in London  
☐ Kwang-on becomes emperor of China  
☐ Napoleon Bonaparte crosses the Alps and invades Italy  
☐ American Express is founded by Henry Wells & William Fargo

**Question 3** What is the capital of Ireland?

☐ Dhaka ☐ Dhaka ☐ Dakar  
☐ Dili ☐ Dili ☐ Djibouti

**Question 4** What is the capital of Thailand?

☐ Beijing ☐ Bangor ☐ Bangkok ☐ Beirut ☐ Berlin

● For your examination, preferably print documents compiled from `automultiple-choice`. ●

●  ● +4/1/52+

AMC SAMPLE TEST

For this test, package `automultiplechoice` is used without any option. Page markers are printed in view of an automated marking from papers scans. DRAFT indications can be cancelled using `\nocancelmark` option.

Commands from `automultiplechoice` are used to print, for each student, two geography questions and two history questions, at random. Questions and answers are shuffled.

**Question 1** Which of the following events are taking place during the year 1971?

☐ The Soviet Union launches Salyut 1  
☐ Apollo 14 lands on the Moon  
☐ Death of Louis Armstrong  
☐ The first commercial Concorde flight takes off

**Question 2** What is the capital of Egypt?

☐ Caracas  
☐ Cayenne  
☐ Cuito  
☐ Conakry  
☐ Chisinau

**Question 3** Which of the following events are taking place during the year 1850?

☐ American Express is founded by Henry Wells & William Fargo  
☐ Napoleon Bonaparte crosses the Alps and invades Italy  
☐ First horse-drawn omnibuses established in London  
☐ Kwang-on becomes emperor of China

**Question 4** What is the capital of Ireland?

☐ Djibouti ☐ Dhaka ☐ Dakar  
☐ Dili ☐ Dili ☐ Dublin

● For your examination, preferably print documents compiled from `automultiple-choice`. ●

# First pages from L<sup>A</sup>T<sub>E</sub>X source detailed in section 2.2 – see sample-separate.pdf

•  • +1/1/60+

AMC SAMPLE TEST

For this test, package `automultiplechoice` is used with `separateanswersheet` option, so that all answers are to be filled on a separate sheet by students. Page markers are printed in view of an automated marking from papers scans. DRAFT indications can be cancelled using `nowatermark` option.

Commands from `automultiplechoice` are used to print, for each student, two geography questions and two history questions, at random. Questions and answers are shuffled.

**Question 1** Which of the following events are taking place during the year 1971?

☐ A The Soviet Union launches Salyut 1  
☐ B The first commercial Concorde flight takes off  
☐ C Death of Louis Armstrong  
☐ D Apollo 14 lands on the Moon

**Question 2** What is the capital of Egypt?

☐ A Cayenne  
☐ B Caracas  
☐ C Cairo  
☐ D Cuiabá  
☐ E Chisinau

**Question 3** Which of the following events are taking place during the year 1850?

☐ A Napoleon Bonaparte crosses the Alps and invades Italy  
☐ B First horse-drawn omnibuses established in London  
☐ C American Express is founded by Henry Wells & William Fargo  
☐ D Kwang-sun becomes emperor of China

**Question 4** What is the capital of Ghana?

☐ A Accra ☐ B Addis Ababa ☐ C Ankara ☐ D Apia

• For your examination, preferably print documents compiled from `automultiple-choice`. •

•  • +1/2/60+

This is the answer sheet: all answers are to be ticked on this page to be taken into account.

**Question 1:** ☐ A ☐ B ☐ C ☐ D  
**Question 2:** ☐ A ☐ B ☐ C ☐ D  
**Question 3:** ☐ A ☐ B ☐ C ☐ D  
**Question 4:** ☐ A ☐ B ☐ C ☐ D

• For your examination, preferably print documents compiled from `automultiple-choice`. •

•  • +2/1/60+

AMC SAMPLE TEST

For this test, package `automultiplechoice` is used with `separateanswersheet` option, so that all answers are to be filled on a separate sheet by students. Page markers are printed in view of an automated marking from papers scans. DRAFT indications can be cancelled using `nowatermark` option.

Commands from `automultiplechoice` are used to print, for each student, two geography questions and two history questions, at random. Questions and answers are shuffled.

**Question 1** Which of the following events are taking place during the year 1901?

☐ A The first line of the Paris Métro is opened  
☐ B Official end of the Cato War of Yunán  
☐ C King George of Greece becomes absolute monarch of Crete  
☐ D Funeral of Queen Victoria in London

**Question 2** What is the capital of Ireland?

☐ A Djibouti ☐ B Dhaka ☐ C Doha  
☐ D Dublin ☐ E Dili ☐ F Dakar

**Question 3** What is the capital of Ghana?

☐ A Apia ☐ B Accra ☐ C Addis Ababa ☐ D Ankara

**Question 4** Which of the following events are taking place during the year 1850?

☐ A Napoleon Bonaparte crosses the Alps and invades Italy  
☐ B First horse-drawn omnibuses established in London  
☐ C American Express is founded by Henry Wells & William Fargo  
☐ D Kwang-sun becomes emperor of China

• For your examination, preferably print documents compiled from `automultiple-choice`. •

•  • +2/2/60+

This is the answer sheet: all answers are to be ticked on this page to be taken into account.

**Question 1:** ☐ A ☐ B ☐ C ☐ D  
**Question 2:** ☐ A ☐ B ☐ C ☐ D  
**Question 3:** ☐ A ☐ B ☐ C ☐ D  
**Question 4:** ☐ A ☐ B ☐ C ☐ D

• For your examination, preferably print documents compiled from `automultiple-choice`. •



First pages from L<sup>A</sup>T<sub>E</sub>X source detailed in section 2.3 – see sample-plain.pdf


AMC

SAMPLE TEST

For this test, package `automultiplechoice` is used with the following options:

- `nopage`, so that no page markers are printed: nothing is planned for future automated marking from papers scans.
- `indivanswers`, so that correct answers are indicated (this is the corrected answer sheet. Without this option, you get the question sheet).


Commands from `automultiplechoice` are used to print, for each student, two geography questions and two history questions, at random. Questions and answers are shuffled.

**Question 1**  Which of the following events are taking place during the year 1971?

☒ The Soviet Union launches Soyuz 1  
☐ The first commercial Concorde flight takes off  
☒ Death of Louis Armstrong  
☒ Apollo 14 lands on the Moon

**Question 2** What is the capital of Egypt?

☐ Cayenne  
☐ Caracas  
☒ Cairo  
☐ Cosenza  
☐ Chisinau

**Question 3**  Which of the following events are taking place during the year 1850?

☐ Napoleon Bonaparte crosses the Alps and invades Italy  
☐ First horse-drawn omnibuses established in London  
☒ American Express is founded by Henry Wells & William Fargo  
☐ Kwang-su becomes emperor of China

**Question 4** What is the capital of Ghana?

☒ Accra ☐ Addis Ababa ☐ Ankara ☐ Apia

1


AMC

SAMPLE TEST

For this test, package `automultiplechoice` is used with the following options:

- `nopage`, so that no page markers are printed: nothing is planned for future automated marking from papers scans.
- `indivanswers`, so that correct answers are indicated (this is the corrected answer sheet. Without this option, you get the question sheet).

Commands from `automultiplechoice` are used to print, for each student, two geography questions and two history questions, at random. Questions and answers are shuffled.

**Question 1**  Which of the following events are taking place during the year 1901?


☐ The first line of the Paris Metro is opened  
☒ Official end of the Cretan War of Vasilata  
☐ King George of Greece becomes absolute monarch of Crete  
☒ Funeral of Queen Victoria in London

**Question 2** What is the capital of Ireland?

☐ Djibouti ☐ Dhaka ☐ Doha  
☒ Dublin ☐ Dili ☐ Dakar

**Question 3** What is the capital of Ghana?

☐ Apia ☒ Accra ☐ Addis Ababa ☐ Ankara

**Question 4**  Which of the following events are taking place during the year 1850?

☐ Napoleon Bonaparte crosses the Alps and invades Italy  
☐ First horse-drawn omnibuses established in London  
☒ American Express is founded by Henry Wells & William Fargo  
☐ Kwang-su becomes emperor of China

1


AMC

SAMPLE TEST


For this test, package `automultiplechoice` is used with the following options:

- `nopage`, so that no page markers are printed: nothing is planned for future automated marking from papers scans.
- `indivanswers`, so that correct answers are indicated (this is the corrected answer sheet. Without this option, you get the question sheet).

Commands from `automultiplechoice` are used to print, for each student, two geography questions and two history questions, at random. Questions and answers are shuffled.

**Question 1**  Which of the following events are taking place during the year 1971?

☐ The first commercial Concorde flight takes off  
☒ Apollo 14 lands on the Moon  
☒ The Soviet Union launches Soyuz 1  
☒ Death of Louis Armstrong

**Question 2**  Which of the following events are taking place during the year 1850?

☐ First horse-drawn omnibuses established in London  
☐ Kwang-su becomes emperor of China  
☐ Napoleon Bonaparte crosses the Alps and invades Italy  
☒ American Express is founded by Henry Wells & William Fargo

**Question 3** What is the capital of Iceland?

☐ Dhaka ☐ Doha ☐ Dakar  
☐ Dili ☒ Dublin ☐ Djibouti

**Question 4** What is the capital of Thailand?

☐ Beijing ☐ Banjul ☒ Bangkok ☐ Beirut ☐ Berlin

1


AMC

SAMPLE TEST

For this test, package `automultiplechoice` is used with the following options:

- `nopage`, so that no page markers are printed: nothing is planned for future automated marking from papers scans.
- `indivanswers`, so that correct answers are indicated (this is the corrected answer sheet. Without this option, you get the question sheet).


Commands from `automultiplechoice` are used to print, for each student, two geography questions and two history questions, at random. Questions and answers are shuffled.

**Question 1**  Which of the following events are taking place during the year 1971?

☒ The Soviet Union launches Soyuz 1  
☒ Apollo 14 lands on the Moon  
☒ Death of Louis Armstrong  
☐ The first commercial Concorde flight takes off

**Question 2** What is the capital of Egypt?

☐ Caracas  
☐ Cayenne  
☒ Cairo  
☐ Cosenza  
☐ Chisinau

**Question 3**  Which of the following events are taking place during the year 1850?

☒ American Express is founded by Henry Wells & William Fargo  
☐ Napoleon Bonaparte crosses the Alps and invades Italy  
☐ First horse-drawn omnibuses established in London  
☐ Kwang-su becomes emperor of China

**Question 4** What is the capital of Iceland?

☐ Djibouti ☐ Dhaka ☐ Dakar  
☐ Dili ☐ Doha ☒ Dublin

1

## 3 Usage

### 3.1 Package options

The following options are available for package `automultiplechoice`:

`noshuffle` cancels answers shuffling for all questions.

`noshufflegroups` cancels groups shuffling.

`answers` produces a common corrected answers sheet.

`indivanswers` shows the boxes that corresponds to correct choices on the question sheet.

`box` includes every question in a  $\text{\LaTeX}$  box, so that they can't be cutted on two different pages.

`asbox` does the same for questions in the separate answer sheet.

`separateanswersheet` asks for a separate answer sheet (see section 2.2 for an example). Commands `\AMCformBegin` and `\AMCform` must be used to describe the separate answer sheet (see section 3.6).

`digits` puts digits instead of letters in the boxes, when `separateanswersheet` (or `insidebox`) is used.

`outsidebox` prints boxes labels outside the boxes on the answersheet when `separateanswersheet` is set.

`init` initializes the random generator from time. *This option is only for testing: don't use it for a real exam!*

`completemulti` adds an answer "None of these answers are correct." at the end of each multiple question (question with no, one or several correct answers), so as to make the difference between "I don't know" and "I think none of the answers are correct".

`insidebox` puts a letter (or a digit if `digits` option is used) inside the boxes, even if `separateanswersheet` is not used. The `insidebox` option is implicitly called when using `separateanswersheet`: no need to call it then.

`calibration` asks for logging positions of boxes and markers in the `.xy` file. Without this option, a  $\text{\LaTeX}$  run updates the document but not the `.xy` file.

`nowatermark` cancels the "DRAFT" indications above pages.

`catalog` is used for formatting a catalog of questions, not an exam. Then the question identifiers will be printed.

`keys` defines the way the question identifiers will be printed on the catalog file. With `keys=next` (the default), the question identifiers will be printed next to the questions numbers. With `keys=line`, the question identifiers will be printed on one line before the question text, so that the question will look close to the final result on the exam copies.

`francais` asks for french localisation.

`lang=XX` asks for localisation in XX language. At present, only CA (Catalan), DE (German), ES (Spanish), FR (French), IT (Italian), JA (Japanese), NO (Norwegian) and NL (Dutch) are available.

`plain` cancels `environ` and `etex` automatic loading. The default behaviour is to load `environ` and `etex` packages if available, as they improve `automultiplechoice`. This is not done when `plain` option is set.

`nopage` cancels markers print and page layout definition (see sample in section 2.3).

`automarks` , when used with `separateanswersheet`, cancels markers print on the subject page (they are only shown on the answer sheet pages).

`postcorrect` tells that correct answers won't be given in the LaTeX source. The teacher will fill one answer sheet for AMC to analyse the scan and set correct answers from it.

`fullgroups` cancels the use of the optional parameter of `\insertgroup` and `\copygroup`, so that each group is always fully inserted and fully copied.

`storebox` asks to use `\storebox` instead of `\savebox` to handle ovals (when using oval shape). The package `storebox` will be loaded.

`pdfform` use this option to produce PDF forms. The PDF sheet won't be printed, but filled by each student with a PDF reader. The completed PDF will then be sent to the teacher, and given to AMC for data capture.

See also section 3.8 for a french version of some of these options.

## 3.2 Questions and answers

We make a difference between two kind of multiple choice questions:

- **Simple questions:** there is one and only one correct choices among the proposed choices, *and this is announced to the student*. Thus, the student is asked to check one answer if he thinks this is the good one, and to check none if he has no idea.
- **Multiple questions:** there can be zero, one or several correct choices among the proposed choices. This is also announced to the student (using the `\multiSymbole` sign, with default ♣), so that the student is asked to check all the boxes corresponding to correct choices, and to let unchecked all boxes corresponding to wrong choices.

`question` Simple questions are enclosed in a `{question}{<id>}` environment, and multiple questions  
`questionmult` are enclosed in a `{questionmult}{<id>}` environment. These environments contain the question text, and the proposed choices inside a `choices`-like environment (see next). The `<id>` argument is a question identifier. Each question must have a unique identifier, different from the other questions identifiers.

---

Questions environments

---

**Question 1** What is the elevation of Mount Everest?

- ☐ 8,253 m  
☐ 8,810 m  
☐ 8,848 m

**Question 2 ♣** Which contries are in the Americas?

- ☐ Cambodia  
☐ Guatemala  
☐ Canada  
☐ Switzerland

```
\begin{question}{everest}
  What is the elevation of
  Mount Everest?
  \begin{choices}
    \correctchoice{8,848\,m}
    \wrongchoice{8,253\,m}
    \wrongchoice{8,810\,m}
  \end{choices}
\end{question}

\begin{questionmult}{americas}
  Which contries are in the Americas?
  \begin{choices}
    \correctchoice{Guatemala}
    \correctchoice{Canada}
    \wrongchoice{Switzerland}
    \wrongchoice{Cambodia}
  \end{choices}
\end{questionmult}
```

`\AMCcompleteMulti`  
`\MCnoCompleteMulti`

For multiple questions, it is sometimes useful to make the difference between a student who thinks that none of he choices are correct, and a student who did not answer the question. The use of package option `completemulti` can be used in this case: it adds a choice to all multiple questions. Commands `\AMCcompleteMulti` and `\AMCnoCompleteMulti` can also be used to change this behaviour for a single question.

---

Additional answer “none” for a single question

---

**Question 3 ♣** Which contries are in the Americas?

- ☐ Guatemala  
☐ Cambodia  
☐ Canada  
☐ Switzerland  
☐ *None of these answers are correct.*

```
\begin{questionmult}{americas}
  \AMCcompleteMulti
  Which contries are in the Americas?
  \begin{choices}
    \correctchoice{Guatemala}
    \correctchoice{Canada}
    \wrongchoice{Switzerland}
    \wrongchoice{Cambodia}
  \end{choices}
\end{questionmult}
```

`choices`  
`choiceshoriz`  
`choicescustom`

Depending on the formatting style for answers, one can choose one of the following ones:

- Environment `choices` is usually chosen for long answers:

The choices environment

**Question 4 ♣** What are the possible uses of latex?

- ☐ Latex is used as a fuel for some space launch vehicles.
- ☐ Latex from the chicle and jelutong trees is used in chewing gum.
- ☐ Natural rubber is the most important product obtained from latex.

```
\begin{questionmult}{latex}
What are the possible uses of latex?
\begin{choices}
\correctchoice{Natural rubber is
the most important product
obtained from latex.}
\correctchoice{Latex from the chicle
and jelutong trees is used in
chewing gum.}
\wrongchoice{Latex is used as a fuel
for some space launch vehicles.}
\end{choices}
\end{questionmult}
```

- environment `choiceshoriz` is chosen for short answers:

The choiceshoriz environment

**Question 5** How many legs for an insect?

- ☐ 2      ☐ 6      ☐ 8

```
\begin{question}{insect}
How many legs for an insect?
\begin{choiceshoriz}
\correctchoice{6}
\wrongchoice{2}
\wrongchoice{8}
\end{choiceshoriz}
\end{question}
```

- environment `choicescustom` is provided to customize answers formatting. See 3.9.3 for details.

`\correctchoice`      As you have seen in these examples, the `choices`-like environments contain `\correctchoice{<text>}` and `\wrongchoice{<text>}` commands, with the text of the proposed choice as argument.

### 3.3 Scoring

`\scoring`      Scoring strategies can be given in the L<sup>A</sup>T<sub>E</sub>X source. They don't have any impact on the question sheet: they are only transmitted to the analysis software through the `.amc` file. See AMC documentation to write proper commands for your needs. `\scoring{<score>}` can be used inside a `question` or `questionmult` environment to describe the scoring strategy for the question, or after a `\correctchoice` or `\wrongchoice` command to describe score associated to a particular choice. `\scoringDefaultM{<score>}` and `\scoringDefaultS{<score>}` define default scoring strategies for multiple and simple questions. `\QuestionIndicative` tags a question that is not taken into account to compute the mark – for example, it can be used for a question about the way students have enjoyed the course.

### 3.4 Groups of questions

Several commands are available that allows shuffling questions for each question sheet. They handle groups of questions. These groups will usually contain questions, but can be made of any L<sup>A</sup>T<sub>E</sub>X content.

`\element`      The command `\element{⟨groupname⟩}{⟨content⟩}` adds element with content `⟨content⟩` to the group named `⟨groupname⟩`. The command `\shufflegroup{⟨groupname⟩}` shuffles elements of group named `⟨groupname⟩`. The command `\insertgroup[⟨n⟩]{⟨groupname⟩}` inserts elements of group `⟨groupname⟩` one after one. If optional parameter `⟨n⟩` is given, only the first `⟨n⟩` elements of the group are inserted in the document. If not, or if `⟨n⟩` is negative, all the elements are inserted. The command `\insertgroupfrom[⟨n⟩]{⟨groupname⟩}{⟨i⟩}` does the same, starting from element at index `⟨i⟩` (the first element has index 0).

As an example without questions in groups elements, let us create a small group named **serie**, containing five elements, and play with it:

Managing groups	
Numbers: one two three four five.	<code>\element{serie}{ one}</code>
Three numbers from the second	<code>\element{serie}{ two}</code>
(index=1) one: two three four.	<code>\element{serie}{ three}</code>
Two of them after shuffling: two four.	<code>\element{serie}{ four}</code>
	<code>\element{serie}{ five}</code>
	<code>Numbers:\insertgroup{serie}.</code>
	 <code>Three numbers from the second (index=1)</code>
	<code>one:\insertgroupfrom[3]{serie}{1}.</code>
	 <code>\shufflegroup{serie}</code>
	<code>Two of them after</code>
	<code>shuffling:\insertgroup[2]{serie}.</code>

`\cleargroup`      The command `\cleargroup{⟨groupname⟩}` clears all the elements of group `⟨groupname⟩`, making an empty group. The command `\copygroup[⟨n⟩]{⟨from⟩}{⟨to⟩}` copies the elements of group `⟨from⟩` to group `⟨to⟩` – if optional parameter `⟨n⟩` is given, only the `⟨n⟩` first elements are copied. If not, or if `⟨n⟩` is negative, all the elements are copied. The command `\copygroupfrom[⟨n⟩]{⟨from⟩}{⟨to⟩}{⟨i⟩}` does the same, starting from element at index `⟨i⟩` (the first element has index 0).

As an example again without questions:

`\cleargroup`  
`\copygroup`  
`\copygroupfrom`

### Copying elements from a group to another

Three digits from 2 to 4 and two letters: A 2 3 F 4.

Three digits and two letters: 2 8 4 E D.

Three digits and two letters: 4 E 2 5 A.

```
\element{digits}{ 1}\element{digits}{ 2}\element{digits}{ 3}
\element{digits}{ 4}\element{digits}{ 5}\element{digits}{ 6}
\element{digits}{ 7}\element{digits}{ 8}\element{digits}{ 9}
\element{letters}{ A}\element{letters}{ B}\element{letters}{ C}
\element{letters}{ D}\element{letters}{ E}\element{letters}{ F}

\shufflegroup{letters}
\cleargroup{mixed}
\copygroupfrom[3]{digits}{mixed}{1}\copygroup[2]{letters}{mixed}
\shufflegroup{mixed}
Three digits from 2 to 4 and two letters:\insertgroup{mixed}.

\shufflegroup{digits}\shufflegroup{letters}
\cleargroup{mixed}
\copygroup[3]{digits}{mixed}\copygroup[2]{letters}{mixed}
\shufflegroup{mixed}
Three digits and two letters:\insertgroup{mixed}.

\shufflegroup{digits}\shufflegroup{letters}
\cleargroup{mixed}
\copygroup[3]{digits}{mixed}\copygroup[2]{letters}{mixed}
\shufflegroup{mixed}
Three digits and two letters:\insertgroup{mixed}.
```

You can find an example involving questions in section 2.

## 3.5 Students identification

There are two ways to associate students to their sheets.

- Always add to one page of each copy some place for the student to write down his name. If you want AMC software to be able to cut the scan around this area to present it to you and ask you to read the written name (this is called manual association), you must use the `\namefield{<descr>}` command. The `<descr>` argument contains the  $\text{\LaTeX}$  code used to format the name field on the page. For example:

### The name field

Name and surname:

.....

```
\namefield{\fbox{
  \begin{minipage}{15em}
    Name and surname:\vspace*{3ex}\par
    \noindent\dotfill\vspace{2mm}
  \end{minipage}
}}
```

You can see that the `\namefield` command has no effect on the produced document. In fact, its only purpose is to log in the `.xy` file information about the position of the name field on the page, to be used by the software analysing the scans.

- For automated student identification, if for example students have a 6-digits student number, you can ask them to code it somewhere on the question sheet. This can be done using the `\AMCcodeGridInt[⟨opts⟩]{⟨key⟩}{⟨ndigits⟩}` command, where `⟨key⟩` is the key identifier, that can be used to retrieve coded student numbers from the scans, and `⟨ndigits⟩` is the number of digits for numbers to be coded.

Student ID	
<div style="display: flex; justify-content: space-between;"> <div style="width: 80%;"> <input type="text"/>0 <input type="text"/>0 <input type="text"/>0 <input type="text"/>0 <input type="text"/>0 <input type="text"/>0  <input type="text"/>1 <input type="text"/>1 <input type="text"/>1 <input type="text"/>1 <input type="text"/>1 <input type="text"/>1  <input type="text"/>2 <input type="text"/>2 <input type="text"/>2 <input type="text"/>2 <input type="text"/>2 <input type="text"/>2  <input type="text"/>3 <input type="text"/>3 <input type="text"/>3 <input type="text"/>3 <input type="text"/>3 <input type="text"/>3  <input type="text"/>4 <input type="text"/>4 <input type="text"/>4 <input type="text"/>4 <input type="text"/>4 <input type="text"/>4  <input type="text"/>5 <input type="text"/>5 <input type="text"/>5 <input type="text"/>5 <input type="text"/>5 <input type="text"/>5  <input type="text"/>6 <input type="text"/>6 <input type="text"/>6 <input type="text"/>6 <input type="text"/>6 <input type="text"/>6  <input type="text"/>7 <input type="text"/>7 <input type="text"/>7 <input type="text"/>7 <input type="text"/>7 <input type="text"/>7  <input type="text"/>8 <input type="text"/>8 <input type="text"/>8 <input type="text"/>8 <input type="text"/>8 <input type="text"/>8  <input type="text"/>9 <input type="text"/>9 <input type="text"/>9 <input type="text"/>9 <input type="text"/>9 <input type="text"/>9 </div> <div style="width: 15%; font-family: monospace; font-size: 0.8em;"> \AMCcodeGridInt{student}{6} </div> </div>	

For smaller number of digits, the “horizontal” form can be preferred:

Student ID, horizontal form	
<div style="display: flex; justify-content: space-between;"> <div style="width: 80%;"> <input type="text"/>0 <input type="text"/>1 <input type="text"/>2 <input type="text"/>3 <input type="text"/>4 <input type="text"/>5 <input type="text"/>6 <input type="text"/>7 <input type="text"/>8 <input type="text"/>9  <input type="text"/>0 <input type="text"/>1 <input type="text"/>2 <input type="text"/>3 <input type="text"/>4 <input type="text"/>5 <input type="text"/>6 <input type="text"/>7 <input type="text"/>8 <input type="text"/>9  <input type="text"/>0 <input type="text"/>1 <input type="text"/>2 <input type="text"/>3 <input type="text"/>4 <input type="text"/>5 <input type="text"/>6 <input type="text"/>7 <input type="text"/>8 <input type="text"/>9 </div> <div style="width: 15%; font-family: monospace; font-size: 0.8em;"> \AMCcodeGridInt[h]{student}{3} </div> </div>	

### 3.6 Separate answer sheet

`\AMCformBegin` To produce separate answer sheets as seen in section 2.2,

`\AMCform`

`\AMCcleardoublepage`

1. use the `separateanswersheet` package option.
2. use the `\AMCformBegin` command at the beginning of the answer sheet description. This command usually follows a command to get a new page. This command can be the classical `\clearpage` for single-sided question sheets, or the `\AMCcleardoublepage` command, that



go to the next odd numbered page, so that the answer sheet is on a separate sheet even when printing in duplex mode.

3. use the `\AMCform` command to insert all boxes for all questions.

See section 2.2 for an example.

### 3.7 Random computation questions

One can use the  $\text{\LaTeX}$  package `fp` to make random computation questions, as can be seen in the following example (don't forget to load package `fp`):

---

Random computation questions

---

**Question 6**      How much are 2 plus 8?

☐ 9      ☒ 10      ☐ 16      ☐ -6

---

```

\begin{question}{simplerandom}
  \FPeval\VQa{trunc(1+random*8,0)}
  \FPeval\VQb{trunc(4+random*5,0)}
  \FPeval\VQsum{clip(VQa+VQb)}
  \FPeval\VQnoA{clip(VQa+VQb-1)}
  \FPeval\VQnoB{clip(VQa*VQb)}
  \FPeval\VQnoC{clip(VQa-VQb)}
  How much are \VQa{} plus \VQb{}?
  \begin{choiceshoriz}
    \correctchoice{\VQsum}
    \wrongchoice{\VQnoA}
    \wrongchoice{\VQnoB}
    \wrongchoice{\VQnoC}
  \end{choiceshoriz}
\end{question}

```

---

In this example, `\VQa` and `\VQb` are used to store two random integers (the first between 1 and 8, and the second between 4 and 8). Then `\VQsum` stores the sum of these two integers, and `\VQnoA`, `\VQnoB` and `\VQnoC` are other values that will be used as distractors in the multiple choice question.

**\AMCIntervals**      In some cases, command `\AMCIntervals{\langle x \rangle}{\langle x0 \rangle}{\langle x1 \rangle}{\langle \delta \rangle}` from `automultiplechoice` can be useful. It adds a sequence of choices made of intervals  $[x_i, x_i + \delta[$  of length  $\langle \delta \rangle$  covering the interval  $[\langle x0 \rangle, \langle x1 \rangle[$ , using `\correctchoice` when  $\langle x \rangle$  lies in the interval, and `\wrongchoice` otherwise.

Pick the right interval

**Question 7** Let  $X$  and  $Y$  be two independent random variables, following exponential laws with respective parameters 5 and 8. In which interval lies the probability  $P[X < Y]$ ?

- |                                     |  |                                     |                                     |                                     |
|-------------------------------------|--|-------------------------------------|-------------------------------------|-------------------------------------|
| <input type="checkbox"/> [0, 0.1[   | <input type="checkbox"/> [0.2, 0.3[            | <input type="checkbox"/> [0.4, 0.5[ | <input type="checkbox"/> [0.6, 0.7[ | <input type="checkbox"/> [0.8, 0.9[ |
| <input type="checkbox"/> [0.1, 0.2[ | <input checked="" type="checkbox"/> [0.3, 0.4[ | <input type="checkbox"/> [0.5, 0.6[ | <input type="checkbox"/> [0.7, 0.8[ | <input type="checkbox"/> [0.9, 1[   |

```
\begin{question}{inf-expo-indep}
  \FPeval\VQa{trunc(2 + random * 4,0)}
  \FPeval\VQb{trunc(6 + random * 5,0)}
  \FPeval\VQr{VQa/(VQa+VQb)}
  Let  $X$  and  $Y$  be two independent random variables, following
  exponential laws with respective parameters  $\VQa$  and  $\VQb$ .
  In which interval lies the probability  $\text{P}[X < Y]$ ?
  \begin{multicols}{5}
    \begin{choices}[o]
      \AMCIntervals{\VQr}{0}{1}{0.1}
    \end{choices}
  \end{multicols}
\end{question}
```

AMCnumericChoices

One can also use the `\AMCnumericChoices` command to ask the student to enter a numerical value as his answer, as in the following example:

Numeric choices

**Question 8**

Compute  $\sqrt{11}$  and round it with two digits after period.

	<input type="checkbox"/> 0	<input type="checkbox"/> 1	<input type="checkbox"/> 2	<input checked="" type="checkbox"/> 3	<input type="checkbox"/> 4	<input type="checkbox"/> 5	<input type="checkbox"/> 6	<input type="checkbox"/> 7	<input type="checkbox"/> 8	<input type="checkbox"/> 9
	.									
<input checked="" type="checkbox"/> +	<input type="checkbox"/> 0	<input type="checkbox"/> 1	<input type="checkbox"/> 2	<input checked="" type="checkbox"/> 3	<input type="checkbox"/> 4	<input type="checkbox"/> 5	<input type="checkbox"/> 6	<input type="checkbox"/> 7	<input type="checkbox"/> 8	<input type="checkbox"/> 9
<input type="checkbox"/> -	<input type="checkbox"/> 0	<input type="checkbox"/> 1	<input checked="" type="checkbox"/> 2	<input type="checkbox"/> 3	<input type="checkbox"/> 4	<input type="checkbox"/> 5	<input type="checkbox"/> 6	<input type="checkbox"/> 7	<input type="checkbox"/> 8	<input type="checkbox"/> 9

```
\begin{questionmultx}{sqrt}
  \FPeval\VQa{trunc(5+random*15,0)}
  \FPeval\VQs{VQa^0.5}

  Compute  $\sqrt{\VQa}$  and round it with two digits after period.

  \AMCnumericChoices{\VQs}{digits=3,decimals=2,sign=true,
    borderwidth=0pt,backgroundcol=lightgray,approx=5}
\end{questionmultx}
```

Note the use of `questionmultx` environment: we need this question to be *multiple* as several boxes has to be ticked, but we can't say that *several answers are correct*, so we don't show the ♣.

Available options that can be used in the second argument of the `\AMCnumericChoices` command are the following (`\bool` can be `true` or `false`, and `\color` must be a color known by the `xcolor` package):

`digits=\num` gives the number of digits to request (defaults to 3).

`decimals=\num` gives the number of digits after period to request (defaults to 0). Note that when `decimals` is positive, the LaTeX package `fp` must be loaded.

`base=\num` gives the base for digits and decimals (defaults to 10).

`significant=\bool` if `true`, the numbers to code are the first *significant* digits from the first argument of `\AMCnumericChoices`. For example, the right answer to `\AMCnumericChoices{56945.23}{digits=2,significant=true}` is 57.

`exponent=\num` gives the number of digits for the exponent, when requesting to enter the result in scientific notation.

`nozero=\bool` if `true`, the choice 0 is removed for all digits. May be useful when `\AMCnumericChoices` is used to get a small ( $< 10$ ) positive value.

`sign=\bool` requests (or not) a signed value (default to `true`).

`exposign=\bool` requests (or not) a signed value of the exponent (default to `true`).

`strict=\bool` if `true`, a box has to be ticked for every digit and for the sign. If `false`, if some digits has no ticked box, they will be set to zero. Defaults to `false`.

`vertical=\bool` if `true`, each digit is represented on one raw. If `false` (default), each digit is represented on one line.

`expovertical=\bool` if `true`, the mantissa is above the exponent. If `false` (default), the mantissa is beside the exponent.

`reverse=\bool` if `true`, place higher values of the digits on the top in vertical mode (defaults to `true`).

`vhead=\bool` if `true`, in vertical mode, a header is placed over all digits rows, made using the command `\AMCncontextVHead` that is originally defined as `\def\AMCncontextVHead#1{\emph{b#1}}`. This default value is useful to number the binary digits. Default value is `false`.

`hspace=\space` sets the horizontal space between boxes (defaults to `.5em`).

`vspace=\space` sets the vertical space between boxes (defaults to `1ex`).

`borderwidth=\space` sets the width of the frame around all the boxes (defaults to `1mm`).

`bordercol=\color` sets the color of the frame (defaults to `lightgray`).

`backgroundcol=\color` sets the background color (defaults to `white`).

`Tsign=\text` sets the text to print at the top of the boxes to set the sign (Can also be redefined by `\def\AMCncontextSign{\text}`), and defaults to be empty).

**Tpoint**= $\langle text \rangle$  sets the text for the period. Can also be redefined by `\def\AMCdecimalPoint{\langle text \rangle}`, and defaults to `\raisebox{1ex}{\bf .}`.

**Texponent**= $\langle text \rangle$  sets the text before the exponent. Can also be redefined by `\def\AMCexponent{\langle text \rangle}`, and defaults to `\times 10^{\textasciicircum}`.

**scoring**= $\langle bool \rangle$  if **true**, a scoring strategy is given to AMC for this question. Defaults to **true**.

**scoreexact**= $\langle num \rangle$  gives the score for an exact answer (defaults to 2).

**exact**= $\langle num \rangle$  sets the maximal distance to the correct integer value (value without the decimal point) for an answer to be said *exact* and be rewarded to **scoreexact** points (defaults to 0).

**scoreapprox**= $\langle num \rangle$  gives the score for an approximative answer (defaults to 1).

**approx**= $\langle num \rangle$  sets the maximal distance to the correct integer value (value without the decimal point) for an answer to be said *approximative* and be rewarded to **scoreapprox** points (defaults to 0).

**scorewrong**= $\langle num \rangle$  gives the score for a wrong answer (defaults to 0).

**ignoreblank** can be used (only with number base 10) to ignore digits for which no box has been ticked. This way, ticking 5 for the first digit, no box for the second and 3 for the third digit will code the number 53, while this would have coded 503 without the **ignoreblank** option (because the default value for the second digit is 0).

**keepas**= $\langle name \rangle$  keeps the value entered by the student in variable  $\{\langle name \rangle\}$ , for future use with *alsocorrect* in another question.

**alsocorrect**= $\langle expression \rangle$  gives another acceptable answer, that can be based on the values entered by the student in the previous questions.

The text added at the end of the questions using `\AMCnumericChoices` when not in the separate answer sheet (and when a separate answer sheet is requested by the `separateanswersheet` package option) can also be set redefining the `\AMCnctextGoto` command, as:

```
\def\AMCnctextGoto{\par{\bf\emph{Please code the answer on
the separate answer sheet.}}}
```

## 3.8 French command names

For backward compatibility, some of `automultiplechoice` commands, environments and package option have their French counterpart. You can always use either the English command or the French equivalent. See table 1 for details.

## 3.9 Customisation

### 3.9.1 Boxes

**\AMCboxStyle** The command `\AMCboxStyle{\langle style \rangle}` can be used to specify the shape, color and dimensions of the boxes to be ticked. The argument  $\langle style \rangle$  is a coma-separated list of  $\langle key \rangle = \langle value \rangle$  pairs, with the following possible  $\langle key \rangle$ s:

type	English	French
command	<code>\namefield</code>	<code>\champnom</code>
environment	<code>choices</code>	<code>reponses</code>
environment	<code>choiceshoriz</code>	<code>reponseshoriz</code>
environment	<code>choicescustom</code>	<code>reponsesperso</code>
command	<code>\correctchoice</code>	<code>\bonne</code>
command	<code>\wrongchoice</code>	<code>\mauvaise</code>
command	<code>\lastchoices</code>	<code>\alafin</code>
command	<code>\AMCIntervals</code>	<code>\choixIntervalles</code>
command	<code>\scoring</code>	<code>\bareme</code>
command	<code>\scoringDefaultM</code>	<code>\baremeDefautM</code>
command	<code>\scoringDefaultS</code>	<code>\baremeDefautS</code>
command	<code>\onecopy</code>	<code>\exemplaire</code>
environment	<code>examcopy</code>	<code>copieexamen</code>
command	<code>\shufflegroup</code>	<code>\melangegroupe</code>
command	<code>\insertgroup</code>	<code>\restituegroupe</code>
command	<code>\AMCform</code>	<code>\formulaire</code>
command	<code>\AMCformBegin</code>	<code>\AMCdebutFormulaire</code>
option	<code>noshuffle</code>	<code>ordre</code>
option	<code>answers</code>	<code>correc</code>
option	<code>indivanswers</code>	<code>correcindiv</code>
option	<code>box</code>	<code>bloc</code>
option	<code>separateanswersheet</code>	<code>ensemble</code>
option	<code>digits</code>	<code>chiffres</code>

Table 1: French equivalent commands

`shape` for the shape to be used: either `square` or `oval`. Note that if `oval` is used, the L<sup>A</sup>T<sub>E</sub>X package `tikz` must be loaded.

`width` for the width of the boxes.

`height` for the height of the boxes.

`size` for the size of the boxes (sets `width` and `height`).

`down` for the length the boxes are to be moved down.

`rule` for the rule width.

`outsidesep` for the distance between the box and the letter when printed outside the box.

`color` for the color (only the box that are to be filled by the students and will be used for data capture). Use something that will be understood by the `xcolor` package.

Default values are

```
\AMCboxStyle{shape=square,size=2.5ex,down=.4ex,
rule=.5pt,outsidesep=.1em,color=black}
```

Setting the box color allows to print the boxes with some color that won't disturb too much the data capture (for example red, but some light grey can also be considered).

Boxes styling		
<b>Question 9</b>	$2 + 2 =$	<code>\AMCboxStyle{shape=oval,color=red}</code>
<b>(A)</b>	1	<code>\begin{question}{sum}\$2+2={}\$</code>
<b>(B)</b>	4	<code>\begin{choiceshoriz}[o]</code>
<b>(C)</b>	10	<code>\wrongchoice{1}\correctchoice{4}</code>
		<code>\wrongchoice{10}</code>
		<code>\end{choiceshoriz}</code>
		<code>\end{question}</code>

### 3.9.2 Codes

One may adapt the codes rendering from `\AMCcodeGrid` to one's needs modifying the following lengths:

- `\AMCcodeHspace` is the amount of horizontal space between two columns of digits,
- `\AMCcodeVspace` is the amount of vertical space between two rows of digits,

Default values are `\AMCcodeHspace=.5em` `\AMCcodeVspace=.5em`

### 3.9.3 Answers

Environment `choicescustom` will make use of the three commands `\AMCbeginAnswer` (before the first answer), `\AMCendAnswer` (after the last answer) and `\AMCanswer{\<box>}{\<text>}` (for each answer) to format the answers. Redefining them properly, some different answers formatting can be achieved. However, this does not seem to work with non-trivial settings...

The choicescustom environment		
<b>Question 10</b>	$2+2=$	$\left( \begin{array}{ccc} \square & 4 & \square & 3 & \square & 2 \end{array} \right)$
<pre> \begin{question}{add} \def\AMCbeginAnswer{\\$ \Big(\$} \def\AMCendAnswer{\\$ \Big)\$} \def\AMCanswer#1#2{#1 #2\hfill} 2+2= \begin{choicescustom} \correctchoice{4} \wrongchoice{2} \wrongchoice{3} \end{choicescustom} \end{question} </pre>		

## 4 Implementation

This package uses the following other packages:

- 1 `\RequirePackage{xcolor}` % `\fcolorbox` to fill (or not) a box
- 2 `\RequirePackage{fancyhdr}` % `\pagestyle{empty}`

```

3 \RequirePackage{bophook} % \AtBeginPage
4 \RequirePackage{xkeyval} % \setkeys
5 \RequirePackage{rotating} % \rotatebox
6 \RequirePackage{fancybox} % \boxput
7 \RequirePackage{expl3}
8 \RequirePackage{csvsimple}
9 \RequirePackage{environ}
10 % \end{macrocode}
11 %
12 % First, we read the options that can be given by AMC through the
13 % |jobname-config.tex| file:
14 % \begin{macrocode}
15 \InputIfFileExists{\jobname-config.tex}%
16 {message>Loading configuration file...^^J}{-}

```

`\AMC@amclog` Informations about questions and choices will be logged to a file with extension `amc`, to be parsed later. Macro `\AMC@amclog` writes to this file.

```

17 \newwrite\AMC@logfile
18 \immediate\openout\AMC@logfile=\jobname.amc
19 \def\AMC@amclog#1{\immediate\write\AMC@logfile{#1}}
20 \def\AMCmessage#1{\AMC@amclog{\string\message{#1}}}

```

`\AMC@LR` Colours management can be faulty in right-to-left mode: in these situations, we will make use of `\LR` from package `bidi` to get back to left-to-right mode. `\AMC@LR` is `\LR` if `bidi` is loaded.

```

21 \AtBeginDocument{\@ifpackageloaded{bidi}{%
22   \PackageInfo{automultiplechoice}{Package bidi loaded: using LR for boxes.}%
23   \let\AMC@LR=\LR}%
24 {\let\AMC@LR=\relax}}%

```

## 4.1 Variables

Counters and boolean variables defined here are internal and should not be modified by the user.

The package defines the following counters:

`\AMCload@counter` number of choices already loaded for current question.

`\AMCid@quest` current question ID number (see section 4.7).

`\AMCid@etud` current student sheet number.

`\AMCid@etudstart` starting student sheet number of the current `onecopy` bloc.

`\AMCid@check` current page checking number.

`\AMCid@etudfin` last student sheet number for the exam.

`\AMCnum@copies` number of exam sheets to produce.

It also defines the following switches:

`\ifAMC@ordre` if choices are never to be shuffled.

`\ifAMC@shuffleG` if groups shuffling is allowed.

`\ifAMC@fullGroups` if groups are always fully inserted by `\insertgroup` and fully copied by `\copygroup`, irrespective to the optional parameter.

`\ifAMC@correthead` if some correction header is to be printed at the beginning.

`\ifAMC@affichekeys` if questions keys are to be printed.

`\ifAMC@keysline` if questions keys should be printed on a single line before the question text.

`\ifAMC@correc` if correct choices are to be checked on the produced document.

`\ifAMC@qbloc` if questions are to be included in  $\text{\LaTeX}$  boxes (so that they can't be splitted on two different pages).

`\ifAMC@asqbloc` if questions are to be included in  $\text{\LaTeX}$  boxes in the answer sheet (so that they can't be splitted on two different pages).

`\ifAMC@rbloc` if answers are to be included in  $\text{\LaTeX}$  boxes (so that they can't be splitted on two different columns for example).

`\ifAMC@textPos` if questions and answers positions are to be logged.

`\ifAMC@extractOnly` if the PDF is only built to extract questions and answers images.

`\ifAMCcomplete@multi` if a choice "None of these answers are correct." is to be added to every multiple question.

`\ifAMCquestionNumber` if AMC should step up the question number for each new question.

`\ifAMC@calibration` if this  $\text{\LaTeX}$  run is used to get page layouts.

`\ifAMC@plain` if `automultiplechoice` won't try to load useful packages (`etex`, `environ`) that extend `automultiplechoice` capabilities.

`\ifAMCune@bonne` if there is at least one correct answer for the current question.

`\ifAMCtype@multi` if the current question is a multiple question.

`\ifAMC@watermark` if the document is a draft, not to be used for exam.

`\ifAMC@ensemble` if answers are to be given on a separate answers sheet.

`\ifAMC@inside@box` if a letter or digit is to be printed inside all boxes.

`\ifAMC@inside@digit` if digits are to be written inside boxes instead of letters (when using a separate answer sheet for example).

`\ifAMC@outside@box` if labels for boxes are to be printed outside the box on the answer sheet.

`\ifAMCformulaire@dedans` is true for questions inside separate answer sheet.

`\ifAMC@zoneformulaire` is true for codes (made by `\AMCcodeGrid`) inside separate answer sheet.

`\ifAMC@pagelayout` is true if the AMC page layout, with signs for scan analysis, is to be used.

`\ifAMC@postcorrect` corresponds to the use of the `postcorrect` package option.

`\ifAMC@automarks` corresponds to the use of the `automarks` package option.



`\ifAMC@invisible` is true is the DVI/PDF output is not important (used for example for scoring strategy extraction).

`\ifAMC@pdfform` is true if the output is a PDF form. This PDF will not be printed but will be filled by the students with a PDF reader and sent back to the teacher.

```

25 \newcount\AMCload@counter
26 \newcount\AMCid@quest\AMCid@quest=-1
27 \newcount\AMCid@check
28 \newcount\AMCid@etud\AMCid@etud=0
29 \newcount\AMCid@etudstart\AMCid@etudstart=0
30 \newcount\AMCid@etudfin
31 \newcount\AMCnum@copies

32 \newif\ifAMC@ordre\AMC@ordrefalse
33 \newif\ifAMC@shuffleG\AMC@shuffleGtrue
34 \newif\ifAMC@fullGroups\AMC@fullGroupsfalse
35 \newif\ifAMC@correthead\AMC@corretheadfalse
36 \newif\ifAMC@affichekeys\AMC@affichekeysfalse
37 \newif\ifAMC@keyslines\AMC@keyslinesfalse
38 \newif\ifAMC@correc\AMC@correcfalse
39 \newif\ifAMC@textPos\AMC@textPosfalse
40 \newif\ifAMC@extractOnly\AMC@extractOnlyfalse
41 \newif\ifAMC@qbloc\AMC@qblocfalse
42 \newif\ifAMC@asqbloc\AMC@asqblocfalse
43 \newif\ifAMC@rbloc\AMC@rblocfalse
44 \newif\ifAMC@complete@multi\AMC@complete@multifalse
45 \newif\ifAMC@questionNumber\AMC@questionNumbertrue
46 \newif\ifAMC@calibration\AMC@calibrationfalse
47 \newif\ifAMC@catalog\AMC@catalogfalse
48 \newif\ifAMC@plain\AMC@plainfalse
49 \newif\ifAMC@bonne
50 \newif\ifAMC@type@multi
51 \newif\ifAMC@watermark\AMC@watermarktrue
52 \newif\ifAMC@inside@box\AMC@inside@boxfalse
53 \newif\ifAMC@outside@box\AMC@outside@boxfalse
54 \newif\ifAMC@ensemble\AMC@ensemblefalse
55 \newif\ifAMC@inside@digit\AMC@inside@digitfalse
56 \newif\ifAMC@formulaire@dedans\AMC@formulaire@dedansfalse
57 \newif\ifAMC@zoneformulaire
58 \newif\ifAMC@pagelayout\AMC@pagelayouttrue
59 \newif\ifAMC@postcorrect\AMC@postcorrectfalse
60 \newif\ifAMC@automarks\AMC@automarksfalse
61 \newif\ifAMC@invisible\AMC@invisiblefalse
62 \newif\ifAMC@pdfform\AMC@pdfformfalse
63 \let\AMCcompleteMulti=\AMCcomplete@multittrue
64 \let\AMCnoCompleteMulti=\AMCcomplete@multifalse

```

`\AMCid@name` The package also defines command `\AMCid@name` to be the current question identifier key.

```
65 \def\AMCid@name{}
```

## 4.2 Dimensions

`\AMCformVSpace` The following dimensions can be modified by the user to adjust questions formatting:

`\AMCformHSpace`

`\AMCinterIrep`

`\AMCinterBrep`

`\AMCformVSpace` is the amount of vertical space between two questions in a separate answer sheet.

`\AMCformHSpace` is the amount of horizontal space between two answers boxes in a separate answer sheet.

`\AMCinterIrep` is the amount of vertical space to be added between two answers.

`\AMCinterBrep` is the amount of vertical space between two boxed answers (see `\AMCBoxedAnswers` and `\ifAMC@rbloc`).

`\AMCinterIquest` is the amount of vertical space left after a question, in standard mode (without package option `box`).

`\AMCinterBquest` is the amount of vertical space left after a question, in 'boxed' mode (with package option `box`).

`\AMCpostOquest` is the amount of vertical space left after an open question.

```
66 \newdimen\AMCformVSpace\AMCformVSpace=1.2ex
67 \newdimen\AMCformHSpace\AMCformHSpace=.3em
68 \newdimen\AMCinterIrep\AMCinterIrep=\z@
69 \newdimen\AMCinterBrep\AMCinterBrep=.5ex
70 \newdimen\AMCinterIquest\AMCinterIquest=\z@
71 \newdimen\AMCinterBquest\AMCinterBquest=3ex
72 \newdimen\AMCpostOquest\AMCpostOquest=7mm
```

### 4.3 Human readable sheet ID position

`\AMCidsPosition` The position of the human readable sheet ID, near the corresponding binary boxes, is set with the `\AMCidsPosition` command, in the form `\AMCidsPosition{pos=<position>,width=<width>,height=<height>}`, where *<position>* is one of `side` (default), `top` and `none`, *<width>* is the width of the box enclosing the ID (default value is 4cm), and *<height>* is the height of the box enclosing the ID (default value is 3ex).

```
73 \newif\ifAMCids@top
74 \newif\ifAMCids@side
75 \newdimen\AMCids@width
76 \newdimen\AMCids@height
77 \define@choicekey*{AMCids}{pos}{\AMCidsVar\AMCidsVarN}{none,top,side}{%
78   \ifcase\AMCidsVarN\relax
79     \AMCids@topfalse\AMCids@sidefalse
80   \or
81     \AMCids@toptrue\AMCids@sidefalse
82   \or
83     \AMCids@topfalse\AMCids@sidesettrue
84   \fi
85 }
86 \define@key{AMCids}{width}{\AMCids@width=#1}
87 \define@key{AMCids}{height}{\AMCids@height=#1}
88 \def\AMCidsPosition#1{\setkeys{AMCids}{#1}}
89 \AMCidsPosition{pos=side,width=4cm,height=3ex}
```

## 4.4 Localisation

In this section, some localised strings or commands are defined, for English, French and Spanish languages.

`\AMCtext` To modify these texts, you can use command `\AMCtext`. For example, `\AMCtext{draft}{\langle text \rangle}` sets the text to be printed behind each page of a draft exam.

```
90 \def\AMCtext#1#2{\expandafter\def\csname AMC@loc@#1\endcsname{#2}}
91 \def\AMClocalized#1{\csname AMC@loc@#1\endcsname}
```

### 4.4.1 English

Text indicating draft exams:

```
92 \def\AMC@loc@draft{DRAFT}
```

Message at page bottom when compiled out of AMC gui:

```
93 \def\AMC@loc@message{For your examination, preferably print
94 documents compiled from auto-multiple-choice.}
```

Announcing a question in a separate sheet (parameter #1 is the question number):

```
95 \def\AMC@loc@qf#1{\textbf{Question #1:}}
```

Announcing a question (parameter #1 is the question number and parameter #2 can be the multiple question symbol, or be empty):

```
96 \def\AMC@loc@q#1#2{\textbf{Question #1} #2}
```

Headers for corrected version and catalog:

```
97 \def\AMC@loc@corrected{Corrected}
98 \def\AMC@loc@catalog{Catalog}
```

Localization text for Explanation

```
99 \def\AMC@loc@explain{\textit{\textbf{Explanation: }}}}
```

Last choice added at the end for multiple questions when option `completemulti` is used:

```
100 \def\AMC@loc@none{None of these answers are correct.}
```

Word for 'question', singular and plural forms:

```
101 \def\AMC@loc@question{question}
102 \def\AMC@loc@questions{questions}
```

Default text to write in the students' name box:

```
103 \def\AMC@loc@namesurname{Name and surname:}
```

### 4.4.2 Catalan

Catalan localisation is called with option `lang=CA`.

```
104 \def\AMC@loc@CA{
105   \def\AMC@loc@draft{PROJECTE}
106   \def\AMC@loc@message{Pel vostre examen, imprimiu preferiblement
107     els documents compilats amb l'ajuda de auto-multiple-choice.}
108   \def\AMC@loc@qf##1{\textbf{Pregunta ##1 :}}
109   \def\AMC@loc@q##1##2{\textbf{Pregunta ##1} ##2}
110   \def\AMC@loc@corrected{Correcci\'}o}
111   \def\AMC@loc@catalog{Cat\'aleg}
112   \def\AMC@loc@explain{\textit{\textbf{Explicaci\'}o : }}}
113   \def\AMC@loc@none{Cap de les respostes \'es correcte.}
```

```

114 \def\AMC@loc@question{pregunta}
115 \def\AMC@loc@questions{preguntas}
116 \def\AMC@loc@namesurname{Nom i cognoms:}
117 }

```

#### 4.4.3 Dutch

Dutch localisation is called with option lang=NL.

```

118 \def\AMC@loc@NL{
119 \def\AMC@loc@draft{Ontwerp}
120 \def\AMC@loc@message{Gebruik bij uw proefwerk bij voorkeur die
121 documenten welke door auto-multiple-choice zijn aangemaakt.}
122 \def\AMC@loc@qf##1{\textbf{Vraag ##1 :}}
123 \def\AMC@loc@q##1##2{\textbf{Vraag ##1} ##2}
124 \def\AMC@loc@corrected{Correctie}
125 \def\AMC@loc@catalog{Catalogus}
126 \def\AMC@loc@none{Geen van de antwoorden is juist.}
127 \def\AMC@loc@question{vraag}
128 \def\AMC@loc@questions{vragen}
129 \def\AMC@loc@namesurname{Achternaam en voornaam:}
130 }

```

#### 4.4.4 French

French localisation is called with option francais, or lang=FR.

```

131 \def\AMC@loc@FR{
132 \def\AMC@loc@draft{PROJET}
133 \def\AMC@loc@message{Pour votre examen, imprimez de pr\'ef\'erence
134 les documents compil\'es \'a l'aide de auto-multiple-choice.}
135 \def\AMC@loc@qf##1{\textbf{Question ##1 :}}
136 \def\AMC@loc@q##1##2{\textbf{Question ##1} ##2}
137 \def\AMC@loc@corrected{Correction}
138 \def\AMC@loc@catalog{Catalogue}
139 \def\AMC@loc@explain{\textit{\textbf{Explication : }}}
140 \def\AMC@loc@none{Aucune de ces r\'eponses n'est correcte.}
141 \def\AMC@loc@question{question}
142 \def\AMC@loc@questions{questions}
143 \def\AMC@loc@namesurname{Nom et pr\'enom :}
144 }

```

#### 4.4.5 German

German localisation is called with option lang=DE.

```

145 \def\AMC@loc@DE{
146 \def\AMC@loc@draft{ENTWURF}
147 \def\AMC@loc@message{Benutzen Sie f\'ur Ihre Pr\'ufung bevorzugt Dokumente die mit
148 auto-multiple-choice erstellt wurden.}
149 \def\AMC@loc@qf##1{\textbf{Frage ##1 :}}
150 \def\AMC@loc@q##1##2{\textbf{Frage ##1} ##2}
151 \def\AMC@loc@corrected{Korrektur}
152 \def\AMC@loc@catalog{Katalog}
153 \def\AMC@loc@explain{\textit{\textbf{Erkl\'arung : }}}
154 \def\AMC@loc@none{Keine dieser Antworten ist korrekt.}

```

```

155 \def\AMC@loc@question{Frage}
156 \def\AMC@loc@questions{Fragen}
157 \def\AMC@loc@namesurname{Vor- und Nachname:}
158 }

```

#### 4.4.6 Italian

Italian localisation is called with option `lang=IT`.

```

159 \def\AMC@loc@IT{
160   \def\AMC@loc@draft{BOZZA}
161   \def\AMC@loc@message{Per l'esame, \e preferibile stampare i documenti
162     a partire da auto-multiple-choice.}
163   \def\AMC@loc@qf##1{\textbf{Domanda ##1:}}
164   \def\AMC@loc@q##1##2{\textbf{Domanda ##1} ##2}
165   \def\AMC@loc@corrected{Correzione}
166   \def\AMC@loc@catalog{Catalogo}
167   \def\AMC@loc@none{Nessuna risposta \e giusta.}
168   \def\AMC@loc@question{domanda}
169   \def\AMC@loc@questions{domande}
170   \def\AMC@loc@namesurname{Nome e cognome:}
171 }

```

#### 4.4.7 Norwegian

Norwegian localisation is called with option `lang=NO`.

```

172 \def\AMC@loc@NO{
173   \def\AMC@loc@draft{UTKAST}
174   \def\AMC@loc@message{Det anbefales {\aa} skrive ut dokumentet
175     for gjennomgang \direkte fra auto-multiple-choice.}
176   \def\AMC@loc@qf##1{\textbf{Oppgave ##1 :}}
177   \def\AMC@loc@q##1##2{\textbf{Oppgave ##1} ##2}
178   \def\AMC@loc@corrected{Rettet}
179   \def\AMC@loc@catalog{Katalog}
180   \def\AMC@loc@none{Ingen svar er riktige.}
181   \def\AMC@loc@question{oppgave}
182   \def\AMC@loc@questions{oppgave}
183   \def\AMC@loc@namesurname{Etternavn og fornavn:}
184 }

```

#### 4.4.8 Portuguese

Portuguese localisation is called with option `lang=PT`.

```

185 \def\AMC@loc@PT{
186   \def\AMC@loc@draft{RASCUNHO}
187   \def\AMC@loc@message{Para o seu exame, use preferencialmente documentos compilados do auto-multiple-cho
188   \def\AMC@loc@qf##1{\textbf{Quest\~ao ##1:}}
189   \def\AMC@loc@q##1##2{\textbf{Quest\~ao ##1} ##2}
190   \def\AMC@loc@corrected{Corrigido}
191   \def\AMC@loc@catalog{Cat\~alogo}
192   \def\AMC@loc@explain{\textit{\textbf{Justifique: }}}
193   \def\AMC@loc@none{Nenhuma das respostas apresentadas est\~a correta.}
194   \def\AMC@loc@question{Quest\~ao}
195   \def\AMC@loc@questions{Quest\~oes}

```



## 4.6 Random

### 4.6.1 Random pseudo-generator

The package uses the pseudo-random bit generator from *TuGBoat* 1994, vol 15:1:

```

228 \ifx\AMC@SR\undefined\newcount\AMC@SR\fi
229 \providecommand\AMC@SRconst{2097152}
230 \providecommand\AMC@SRset[1]{\global\AMC@SR#1 \ignorespaces}
231 \providecommand\AMC@SRadvance{%
232   \begingroup%
233     \ifnum\AMC@SR<\AMC@SRconst\relax\AMC@SR@count\z@else\AMC@SR@count\@ne\fi%
234     \ifodd\AMC@SR\advance\AMC@SR@count\@ne\fi%
235     \global\divide\AMC@SR\tw@%
236     \ifodd\AMC@SR@count\global\advance\AMC@SR\AMC@SRconst\relax\fi%
237   \endgroup}
238 \providecommand\AMC@SRbit{\AMC@SRadvance\ifodd\AMC@SR1\else0\fi}
239 \providecommand\AMC@SRtest[2]{\AMC@SRadvance%
240   \ifodd\AMC@SR#2\else#1\fi\ignorespaces}
241 \providecommand\AMC@SRvalue{\number\AMC@SR}

```

`\AMCrandomseed` The seed of this generator is set to 1515, but another value can be given using the command `\AMCrandomseed{<seed>}`.

```

242 \AMC@SRset{1515}
243 \def\AMCrandomseed#1{\AMC@SRset{#1}}

```

### 4.6.2 Uniform random deviates

`\AMC@SRnextByte` This generator is used to build first a 20-bit uniform integer generator (macro `\AMC@SRnextByte`).  
`\AMC@SRmax` Then, using modulo, a (nearly) uniform generator on  $\{0, \dots, n - 1\}$  is built: command `\AMC@SRmax{n}` puts in `\AMC@SR@count` the random deviate.

```

244 \newcount\AMC@SR@count
245 \def\AMC@SR@time{\AMC@SRset{\time}}
246 \newcount\AMC@SRnum
247 \def\AMC@SRnextByte{\AMC@SRnum=\z@%
248   \AMC@SR@count=20%
249   \loop\multiply\AMC@SRnum\tw@%
250     \AMC@SRtest{\advance\AMC@SRnum\@ne}{}%
251   \ifnum\AMC@SR@count>\@ne\advance\AMC@SR@count\m@ne\repeat%
252 }
253 \newcommand\AMC@SRmax[1]{\AMC@SRnextByte%
254   \AMC@SR@count=\AMC@SRnum%
255   \divide\AMC@SR@count by #1\relax%
256   \multiply\AMC@SR@count by #1\relax%
257   \advance\AMC@SRnum by -\AMC@SR@count%
258 }

```

### 4.6.3 Tokens shuffling

`\AMCsw@p` The package defines the macro `\AMCsw@p` to swap the values of two token registers given as parameters.  
`\AMC@shuffletoks`

After defining  $n$  token registers `\foo@i`, `\foo@ii`, `\foo@iii`, `\foo@iv` and so on, you can shuffle them using `\AMC@shuffletoks[<a>]{<n>}{<foo>}`. With optional argument  $\langle a \rangle$ , registers are shuffled from number  $\langle a \rangle$  to  $\langle n \rangle$  (default value for  $\langle a \rangle$  is 1).

```

259 \newcount\AMC@sti
260 \newcount\AMC@stil
261 \newtoks\AMCsw@p@
262 \newcommand\AMCsw@p[2]{%
263   \global\AMCsw@p@=#1%
264   \global#1=#2%
265   \global#2=\AMCsw@p@}
266 \newcommand{\AMC@shuffletoks}[3][\@one]{%
267   \AMC@sti=#2\relax%
268   \AMC@stil=#2\relax%
269   \advance\AMC@stil\@one%
270   \advance\AMC@stil -#1\relax%
271   \@whilenum\AMC@sti>#1\do{%
272     \AMC@SRmax{\AMC@stil}\advance\AMC@SRnum #1\relax%
273     \AMCsw@p{\csname #3\romannumeral\AMC@SRnum\endcsname}%
274     {\csname #3\romannumeral\AMC@sti\endcsname}%
275     \advance\AMC@sti\m@ne\relax%
276     \advance\AMC@stil\m@ne\relax%
277   }}

```

## 4.7 Keys numbering

`\AMC@unnumero` This package allocates a unique integer ID to each question key from the questionnaire.  
`\AMC@affecte` The counter `\AMC@numeration` keeps track of the number of keys which already had an ID. Command `\AMC@definitnumero{n}{key}` allocates ID *n* to the key *key*. Command `\AMC@prepare{key}` looks if an ID had already been associated to *key*, and, if not, makes a new ID allocation for *key*. Command `\AMC@unnumero{key}` returns the ID associated with *key* (creating one if necessary). Command `\AMC@affecte{key}{\cnt}` give to counter `\cnt` the value of the ID associated to *key* (creating one if necessary).

```

278 \newcount\AMC@numeration\AMC@numeration=\z@%
279 \def\AMC@definitnumero#1#2{\AMCmessage{NUM=#1=#2}%
280   \expandafter\global\expandafter\def\csname AMC@numtab@#2\endcsname{#1}}
281 \def\AMC@prepare#1{\expandafter\ifx\csname AMC@numtab@#1\endcsname\relax%
282   \global\advance\AMC@numeration\@one%
283   \expandafter\AMC@definitnumero\expandafter{\the\AMC@numeration}{#1}\fi}
284 \def\AMC@unnumero#1{\AMC@prepare{#1}\csname AMC@numtab@#1\endcsname}
285 \def\AMC@affecte#1#2{\AMC@prepare{#1}\global#2=\csname AMC@numtab@#1\endcsname}

```

## 4.8 Boxes

### 4.8.1 Character logging

`\AMC@logchar` The command `\AMC@logchar{<char>}{<key>}` logs the character written in the box referenced as *<key>* in the .amc file. This is used in catalog mode, to get understandable references to answers from the statistics tables of the ODS export.

```

286 \def\AMC@logchar#1#2{%
287   \protected@write\AMC@logfile{}{%
288     \string\answer%
289     {\the\AMCid@etud/\thepage:#2}%
290     {#1}}%
291 }

```



## 4.8.2 Position logging

`\AMC@tracebox` Command `\AMC@tracebox{<trace>}{<key>}{<content>}` makes a L<sup>A</sup>T<sub>E</sub>X box around `<content>`, and, if `<trace>` is not empty, logs to the .xy file informations to be able to compute exact location of this box on the page, attached to the box identification `<key>`.

Command `\AMC@pagepos` logs page and page size informations at the beginning of each page.

```

292 \def\AMC@shapename@{\ifAMC@invisible none\else\AMC@shapename\fi}
293 \def\AMC@tracepos#1#2{%
294   \ifAMC@calibration\ifx\@empty#1\@empty\else%
295     \pdfsavepos\protected@write\AMC@XYFILE{}{%
296       \string\tracepos%
297       {\the\AMCid@etud/\thepage:#2}%
298       {\noexpand\number\pdflastxpos sp}%
299       {\noexpand\number\pdflastypos sp}%
300       {\AMC@shapename}}%
301   \fi\fi}
302 \def\AMC@traceposx#1#2{%
303   \ifAMC@calibration\ifx\@empty#1\@empty\else%
304     \pdfsavepos\protected@write\AMC@XYFILE{}{%
305       \string\tracepos%
306       {\the\AMCid@etud/\thepage:#2}%
307       {\noexpand\number\pdflastxpos sp}%
308       {0sp}%
309       {\AMC@shapename}}%
310   \fi\fi}
311 \def\AMC@traceposy#1#2{%
312   \ifAMC@calibration\ifx\@empty#1\@empty\else%
313     \pdfsavepos\protected@write\AMC@XYFILE{}{%
314       \string\tracepos%
315       {\the\AMCid@etud/\thepage:#2}%
316       {0sp}%
317       {\noexpand\number\pdflastypos sp}%
318       {\AMC@shapename}}%
319   \fi\fi}
320 \newcommand\AMC@tracebox[3]{%
321   \vbox{\AMC@traceposy{#1}{#2}%
322     \hbox{\AMC@traceposx{#1}{#2}#3\AMC@traceposx{#1}{#2}}%
323     \AMC@traceposy{#1}{#2}}%
324 \def\AMC@pagepos{%
325   \ifAMC@calibration\protected@write\AMC@XYFILE{}{%
326     \string\page%
327     {\the\AMCid@etud/\thepage/\the\AMCid@check}%
328     {\the\paperwidth}{\the\paperheight}%
329     {\the\pdfpagewidth}{\the\pdfpageheight}}\fi}

```

`\AMCdontScan` The commands `\AMCdontScan`, `\AMCdontAnnotate` and `\AMCreTick` write into the xy file instructions related to the current question.

```

\AMCreTick 330 \newcommand{\AMCdontScan}{\ifAMC@calibration\immediate\write\AMC@XYFILE{\string\dontscan{\the\AMCid@etud,
331 \newcommand{\AMCdontAnnotate}{\ifAMC@calibration\immediate\write\AMC@XYFILE{\string\dontannotate{\the\AMC
332 \newcommand{\AMCreTick}{\ifAMC@calibration\immediate\write\AMC@XYFILE{\string\retick{\the\AMCid@etud,\the
333 %

```

`\AMC@tracechar` The macro `\AMC@tracechar{<char>}{<unused>}{<trace>}{<key>}` is used to log (for further pro-

cessing with AMC), into to .xy file, the character used to identify the box.

```

334 \newcommand\AMC@tracechar[4]{%
335   \ifAMC@calibration\ifx\@empty#3\@empty\else%
336     \protected@write\AMC@XYFILE{}\string\xyopen{#1}}%
337     \string\boxchar{\the\AMCid@etud/\thepage:#4}{#1}%
338   }%
339 \fi\fi%
340 }

```

**amcxyfile** The following lines defines an environment to tag positions outputs for a particular part of the document. This is used mainly for documentation or testing.

```

341 \newenvironment{amcxyfile}[1]{%
342   \protected@write\AMC@XYFILE{}\string\xyopen{#1}}%
343 }{%
344   \protected@write\AMC@XYFILE{}\string\close{}%
345 }

```

**\AMCzone** The `\AMCzone[flags]{zone name}{zone content}` is a simple call to `\AMC@tracebox`:

```

346 \newcommand{\AMCzone}[3][\AMC@tracebox]{1}{__zone:#1:#2}{#3}}

```

**\namefield** The `\namefield{name field content}` is a simple call to `\AMCzone`:

```

347 \newcommand{\namefield}[2][id]{\AMCzone[#1]{_n}{#2}}

```

It is used to enclose the page region where students are to write their names, so as to retrieve it easily from the scans.

**\namefielddots** The command `\namefielddots` can be used to fill a line with dots (printed sheets) or use a text field in PDF forms:

```

348 \newcommand{\namefielddots}{%
349   \noindent%
350   \ifAMC@pdfform%
351     \hspace*{\fill}%
352     \TextField[name={\the\AMCid@etud:namefield},width=.95\linewidth,bordercolor=0 0 0]{}%
353     \hspace*{\fill}%
354   \else%
355     \dotfill
356   \fi%
357 }

```

As an example,

```

\namefield{\fbox{%
  \begin{minipage}{5cm}
    Name:

    \vspace*{.5cm}
    \namefielddots
    \vspace{2mm}
  \end{minipage}}}

```

produces the following box:

Name:

.....

and outputs information about the position of the box in the .xy file, as seen in section 5.1.

### 4.8.3 Boxes to be checked by students

There are two styles for boxes to be checked by the students. The first one is an empty box, printed beside the answer. The second is a box with a character in it. It is mainly used when answers are to be given on a separate answer sheet.

These boxes can be drawn using command `\AMC@answerBox@{<char>}{<answer>}{<trace>}{<key>}`: `<char>` is the character to print inside the box, `<trace>` is non-empty if you want to log the box position in the .xy file, `<key>` is the box identification, and `<answer>` is an answer to be written in the box (or `\AMC@checkbox` for filling the box).

Depending on the required shape for the boxes, the corresponding

`\AMC@shape@xxx{<char>}{<answer>}{<trace>}{<key>}`

command is used.

- `\AMC@answerBox@{K}{1}{test}` produce the box K, writing the lines in the .xy file shown in section 5.2.
- `\AMC@answerBox@{K}{\AMC@checkbox}{}` produces
- `\AMC@answerBox@{}{8}{}` produces 8
- `\AMC@answerBox@{K}{8}{1}{testb}` produces 8 with `\AMC@boxStyle{shape=oval,color=red}`

```
358 \def\AMC@checkbox{}
359 \let\AMC@new@savebox=\newsavebox
360 \let\AMC@save@box=\savebox
361 \let\AMC@use@box=\usebox
362 \newif\ifAMC@draw@cross
```

The `\AMC@smashcentered{<text>}` command shows the `<text>` centered at point.

```
363 \newbox\AMC@smashbox
364 \newdimen\AMC@smashboxheight
365 \newcommand{\AMC@smashcentered}[1]{%
366   \setbox\AMC@smashbox\hbox{#1}%
367   \AMC@smashboxheight=\ht\AMC@smashbox%
368   \advance\AMC@smashboxheight by \dp\AMC@smashbox%
369   \vfuzz=\AMC@smashboxheight\hfuzz=\wd\AMC@smashbox%
370   \hspace*{-.5\wd\AMC@smashbox}\hbox to .5\wd\AMC@smashbox{%
371     \vbox to 0pt{%
372       \vspace*{-.5\AMC@smashboxheight}\vbox to .5\AMC@smashboxheight{%
373         \box\AMC@smashbox}}}%
374 }
```

`\AMC@setcolors@{<trace>}{<answer>}` sets colours `\AMC@boxcolor@` and `\AMC@fillcolor@` according to its arguments. It also sets the `\ifAMC@draw@cross` switch if AMC should draw a cross instead of filling the box.

```

375 \newcommand\AMC@setcolors@[2]{%
376   \def\AMC@boxcolor@{\AMC@boxcolor}%
377   \ifx\@empty#1\@empty \def\AMC@boxcolor@{black}\fi%
378   \ifAMC@correc\def\AMC@boxcolor@{black}\fi%
379   \def\AMC@fillcolor@{\ifx #2\AMC@checkedbox%
380     \AMC@boxcolor@\else white\fi}%
381   \AMC@draw@crossfalse%
382   \ifKV@AMCdim@cross\ifx #2\AMC@checkedbox%
383     \AMC@draw@crosstrue\fi\fi%
384 }
385 \newcommand\AMC@answerBox@[4]{%
386   \ifAMC@catalog%
387     \AMC@logchar{#1}{#4}%
388     \fi%
389   \AMC@LR{\hspace{0pt}%
390     \lower\AMC@boxeddown\hbox{\csname AMC@shape@\AMC@shapename@\endcsname%
391       {\AMC@choiceLabelFormat{#1}{#2}{#3}{#4}}}%
392 }
393 \newcommand\AMC@shapeprepare@square{}
394 \newcommand\AMC@shape@square[4]{%
395   \fboxsep=\z@\fboxrule=\AMC@boxedrule%
396   \AMC@setcolors@{#3}{#2}%
397   \ifKV@AMCdim@cross\def\AMC@fillcolor@{white}\fi%
398   \fcolorbox{\AMC@boxcolor@}{\AMC@fillcolor@}%
399   {%
400     \boxput*(0,0){%
401       \ifAMC@draw@cross\AMC@crosschar\fi%
402     }{%
403       \vbox to \AMC@boxedheight{%
404         \AMC@tracepos{#3}{#4}%
405         \vfill%
406         \hbox to \AMC@boxedwidth{\hfill%
407           \AMC@smashcentered{\textcolor{\AMC@boxcolor@}{#1}}%
408           \AMC@smashcentered{#2}%
409           \hfill}\vfill}}%
410       \AMC@tracepos{#3}{#4}}%
411 }
  \AMC@makeovalbox{<trace>}{<answer>}{<box>} prepares an oval frame in the LATEX box <box>.
412 \newcommand\AMC@makeovalbox[3]{%
413   \AMC@setcolors@{#1}{#2}%
414   \ifKV@AMCdim@cross\def\AMC@fillcolor@{white}\fi%
415   \AMC@save@box{#3}{%
416     \begin{tikzpicture}%
417       \useasboundingbox (-0.5\AMC@boxedwidth-0.5\AMC@boxedrule,0.5\AMC@boxedheight+0.5\AMC@boxedrule)
418       rectangle (0.5\AMC@boxedwidth+0.5\AMC@boxedrule,-0.5\AMC@boxedheight-0.5\AMC@boxedrule);
419       \draw[\AMC@boxcolor@,fill=\AMC@fillcolor@,line width=\AMC@boxedrule,rounded corners=\AMC@oval@radius]
420       (-0.5\AMC@boxedwidth,0.5\AMC@boxedheight)
421       rectangle (0.5\AMC@boxedwidth,-0.5\AMC@boxedheight);
422       \ifAMC@draw@cross
423         \draw[\AMC@boxcolor@,line width=\AMC@crossrule]
424         (-0.5\AMC@boxedwidth,0.5\AMC@boxedheight) -- (0.5\AMC@boxedwidth,-0.5\AMC@boxedheight)
425         (0.5\AMC@boxedwidth,0.5\AMC@boxedheight) -- (-0.5\AMC@boxedwidth,-0.5\AMC@boxedheight);
426       \fi

```

```

427 \end{tikzpicture}}%
428 }
429 \newcommand\AMC@shapeprepare@oval{%
430 \ifx\AMC@ovalbox@R\undefined\else%
431 \AMC@makeovalbox{1}{}\AMC@ovalbox@R}%
432 \AMC@makeovalbox{1}{\AMC@checkbox}\AMC@ovalbox@RF}%
433 \AMC@makeovalbox{1}{\AMC@ovalbox@}%
434 \AMC@makeovalbox{1}{\AMC@checkbox}\AMC@ovalbox@F}%
435 \fi%
436 }
437 \newcommand\AMC@shape@oval[4]{%
438 \AMC@setcolors@{#3}{#2}%
439 \AMC@tracebox{#3}{#4}{\boxput*(0,0){%
440 \AMC@smashcentered{\textcolor{\AMC@boxcolor@}{#1}}%
441 \AMC@smashcentered{#2}%
442 }{}}%
443 \ifx\@empty#3\@empty%
444 \ifx #2\AMC@checkbox%
445 \AMC@use@box{\AMC@ovalbox@F}%
446 \else%
447 \AMC@use@box{\AMC@ovalbox@}%
448 \fi%
449 \else%
450 \ifx #2\AMC@checkbox%
451 \AMC@use@box{\AMC@ovalbox@RF}%
452 \else%
453 \AMC@use@box{\AMC@ovalbox@R}%
454 \fi%
455 \fi%
456 }}%
457 }
458 \newcommand\AMC@shapeprepare@form{}
459 \newcommand\AMC@shape@form@base[5]{%
460 \ifx #2\AMC@checkbox%
461 \def\AMC@shape@form@ticked{true}%
462 \else%
463 \def\AMC@shape@form@ticked{false}%
464 \fi%
465 \AMC@tracebox{#3}{#4}{%
466 \CheckBox[checked=\AMC@shape@form@ticked,%
467 checkboxsymbol=\ding{110},name={#5},%
468 bordercolor=0 0 0,%
469 width=\AMC@boxedwidth,height=\AMC@boxedheight]{}{}}%
470 }%
471 }
472 \newcommand\AMC@shape@form[4]{%
473 \AMC@shape@form@base{#1}{#2}{#3}{#4}{\the\AMCid@etud:#4}%
474 }
475 \newcommand\AMC@shapeprepare@none{}
476 \newcommand\AMC@shape@none[4]{ #1 }

```

`\AMC@answerBox` Command `\AMC@answerBox` is the same as `\AMC@answerBox@`, but if `\char` is empty, it is replaced  
`\AMC@choiceLabel` by an arabic or alphabetical counter, depending on the use of the `digits` package option.  
`choiceLabelFormat` To use another way to label the choices boxes, the user can redefine the `\AMC@choiceLabel`

macro, which takes as argument the name of the counter used to number the choices. One can for example use `\def\AMCchoiceLabel#1{\alph{#1}}` to ask for lowercase letters.

To write these labels with another font, size, or so, the user can redefine the `\AMCchoiceLabelFormat` macro, which takes as argument the label. One can for example get sans serif bold labels with `\def\AMCchoiceLabelFormat#1{{\textsf{\textsf{#1}}}}`.

```

477 \def\AMCchoiceLabel#1{%
478   \ifAMC@inside@digit\arabic{#1}%
479   \else\Alph{#1}\fi%
480 }
481 \def\AMCchoiceLabelFormat#1{#1}
482 \newcounter{AMC@ncase}
483 \setcounter{AMC@ncase}{0}
484 \newcommand\AMC@answerBox[4]{%
485   \AMC@answerBox@{\ifx\@empty#1\@empty%
486     \AMCchoiceLabel{AMC@ncase}%
487     \else #1\fi}{#2}{#3}{#4}}

```

`\AMCboxStyle` The dimensions of these box are managed by `\AMCboxDimensions{<sizes>}`, where *<sizes>* is a coma separated list of *<name>=<dimension>* constructs. Here, *<name>* can be *size* for the box size, *rule* for the box rule width, *down* for moving the box down, *color* for the box color and *outsidesep* for the distance between the box and the letter (when outside the box).

The *<color>* value given to *color* is a color that should be defined for the `xcolor` package. This color is used only in the case the box will be used for data capture: it is not used on the corrected answer sheet (`answers` or `indivanswers` package option), and not used on the subject part of an exam with a separate answer sheet (`separateanswersheet` package option).

The `\AMCboxColor{<color>}` command is defined as an alias to `\AMCboxStyle{color=<color>}`, and `\AMCboxDimensions` as an alias to `\AMCboxStyle`, for backward compatibility.

```

488 \newlength\AMC@boxedrule
489 \newlength\AMC@crossrule
490 \newlength\AMC@boxeddown
491 \newlength\AMC@boxedwidth
492 \newlength\AMC@boxedheight
493 \newlength\AMC@oval@radius
494 \newlength\AMC@outside@sep
495 \define@choicekey{AMCdim}{shape}{square,oval,form,none}{\def\AMC@shapename{#1}}
496 \define@key{AMCdim}{size}{\AMC@boxedwidth=#1\AMC@boxedheight=#1}
497 \define@key{AMCdim}{height}{\AMC@boxedheight=#1}
498 \define@key{AMCdim}{width}{\AMC@boxedwidth=#1}
499 \define@key{AMCdim}{rule}{\AMC@boxedrule=#1}
500 \define@key{AMCdim}{outsidesep}{\AMC@outside@sep=#1}
501 \define@key{AMCdim}{down}{\AMC@boxeddown=#1}
502 \define@key{AMCdim}{color}{\def\AMC@boxcolor{#1}}
503 \define@boolkey{AMCdim}{cross}[false]{}
504 \define@key{AMCdim}{crosschar}{\textbf{\textsf{X}}}{\def\AMC@crosschar{#1}}
505 \define@key{AMCdim}{crossrule}[1.5pt]{\AMC@crossrule=#1}
506 \def\AMC@shapeprepare{\csname AMC@shapeprepare@\AMC@shapename@ \endcsname}
507 \def\AMCboxStyle#1{%
508   \setkeys{AMCdim}{#1}%
509   \ifnum\AMC@boxedwidth<\AMC@boxedheight%
510     \AMC@oval@radius=\AMC@boxedwidth\divide\AMC@oval@radius\tw@%
511   \else%
512     \AMC@oval@radius=\AMC@boxedheight\divide\AMC@oval@radius\tw@%

```

```

513 \fi%
514 \AMC@shapeprepare%
515 }
516 \AMCboxStyle{shape=square,size=2.5ex,down=.4ex,rule=.5pt,outsidesep=.1em,color=black,cross,crosschar,crossrule}
517 \newcommand\AMCboxColor[1]{\AMCboxStyle{color=#1}}
518 \let\AMCboxDimensions=\AMCboxStyle

\AMCboxOutsideLetter Command \AMC@box{<char>}{<answer>} prints a box with character <char> inside, showing an-
\AMC@box answer <answer> (\AMC@checkbox to get a filled box), using global variables to identify the box
\AMC@formBox@ (question and choice).
\AMC@formBox@ It calls \AMC@formBox@{<char>}{<answer>}{<trace>}{<key>} to actually render the box.
\AMC@formBox@ Command \AMC@formBox simply sets the first argument when empty before calling \AMC@formBox@.
\AMC@formBox@ The command \AMCboxOutsideLetter{<box>}{<char>} is called to print the box and the
\AMC@formBox@ character <char> outside (and next to) it. The character is formatted using \AMCoutsideLabelFormat
\AMC@formBox@ first: if you need bold characters, redefine it with \def\AMCoutsideLabelFormat#1{\textbf{#1}}
\AMC@formBox@ \AMC@keyBox@ is used instead of \AMC@formBox@ when the text that corresponds to the answer
\AMC@formBox@ is the letter/character inside the box itself (see \AMCcodeGrid and \AMCnumericChoices.

519 \def\AMCoutsideLabelFormat#1{#1}
520 \newcommand\AMCboxOutsideLetter[2]{#1\nobreak\hspace{.1em}\AMCoutsideLabelFormat{#2}}
521 \newif\ifAMC@printformoutside%
522 \newcommand\ifAMC@printformoutside{%
523 \AMC@printformoutside@false%
524 \ifAMC@ensemble\ifAMC@outside@box%
525 \ifAMC@formulaire@dedans\AMC@printformoutside@true\fi%
526 \ifAMC@zoneformulaire\AMC@printformoutside@true\fi%
527 \fi\fi%
528 \ifAMC@printformoutside%
529 }
530 \newcommand\AMC@formBox@[4]{%
531 \ifAMC@printformoutside% letter to be written outside the box
532 \AMCboxOutsideLetter{\AMC@answerBox@{#2}{#3}{#4}}{#1}%
533 \else%
534 \AMC@answerBox@{#1}{#2}{#3}{#4}%
535 \fi%
536 \AMC@tracechar{#1}{#2}{#3}{#4}%
537 }
538 \newif\ifAMC@printkeyoutside%
539 \newcommand\ifAMC@printkeyoutside{%
540 \AMC@printkeyoutside@false%
541 \ifAMC@ensemble%
542 \ifAMC@outside@box\AMC@printkeyoutside@true\fi%
543 \else%
544 \ifAMC@inside@box\else\AMC@printkeyoutside@true\fi%
545 \fi%
546 \ifAMC@printkeyoutside%
547 }
548 \newcommand\AMC@keyBox@[4]{%
549 \ifAMC@printkeyoutside%
550 \AMCboxOutsideLetter{\AMC@answerBox@{#2}{#3}{#4}}{#1}%
551 \else%
552 \AMC@answerBox@{#1}{#2}{#3}{#4}%
553 \fi%

```

```

554 \AMC@tracechar{#1}{#2}{#3}{#4}%
555 }
556 \newcommand\AMC@formBox[4]{%
557 \AMC@formBox@{\ifx\@empty#1\@empty%
558 \AMCchoiceLabel{AMC@ncase}%
559 \else #1\fi}{#2}{#3}{#4}%
560 }
561 \newcommand{\AMC@box}[2]{%
562 \ifAMC@ensemble%
563 \ifAMC@zoneformulaire% for codes inside form sheet
564 \protect\AMC@formBox{#1}{#2}{1}{case:\AMCid@name:\the\AMCid@quest,\the\AMCrep@count}%
565 \else%
566 \ifAMCformulaire@dedans% for answer boxes inside form sheet
567 \protect\AMC@formBox{#1}{#2}{1}{case:\AMCid@name:\the\AMCid@quest,\the\AMCrep@count}%
568 \else% outside form sheet: not to be read during data capture
569 \AMC@formBox{#1}{#2}{1}{casequestion:\AMCid@name:\the\AMCid@quest,\the\AMCrep@count}%
570 \fi\fi%
571 \else% no separate sheet for answers: always read
572 \ifAMC@inside@box%
573 \AMC@formBox{#1}{#2}{1}{case:\AMCid@name:\the\AMCid@quest,\the\AMCrep@count}%
574 \else%
575 \AMC@formBox@{\#2}{1}{case:\AMCid@name:\the\AMCid@quest,\the\AMCrep@count}%
576 \fi%
577 \fi%
578 }

```

#### 4.8.4 Scoring zones

`\AMCscoreZone` The source file can define zones that will be used to print scores when annotating the completed answer sheets. The command `\AMCscoreZone{<zone>}` logs these zones positions on the page.

```

579 \newif\ifAMCsz@logged\AMCsz@loggedfalse
580 \newcommand{\AMCscoreZone}[1]{%
581 \ifAMC@ensemble%
582 \ifAMCformulaire@dedans%
583 \AMC@tracebox{1}{score::\the\AMCid@quest,-1}{#1}%
584 \else%
585 \AMC@tracebox{1}{scorequestion::\the\AMCid@quest,-1}{#1}%
586 \fi%
587 \else%
588 \AMC@tracebox{1}{score::\the\AMCid@quest,-1}{#1}%
589 \fi%
590 \ifAMCsz@logged\else%
591 \AMCmessage{VAR:scorezones=1}%
592 \global\AMCsz@loggedtrue%
593 \fi%
594 }

```

#### 4.8.5 Binary boxes

The package prints on each page some boxes that code (like binary digits) student sheet number, page number and a check number, so as to be read easily from scans after exam.

`\AMCid@checkmax` The check number is just decreased each page. Its maximum value is `\AMCid@checkmax`.  
`\AMC@NCBetud`  
`\AMC@NCBpage`  
`\AMC@NCBcheck`



The number of binary digits used to print student sheet number, page and check number are `\AMC@NCBetud`, `\AMC@NCBpage` and `\AMC@NCBcheck`. The number of the first page is `\AMC@premierecopie`.

The length of zone reserved for binary boxes is `\AMC@CBtaille`.

```
595 \def\AMCid@checkmax{60}
596 \def\AMC@NCBetud{12}
597 \def\AMC@NCBpage{6}
598 \def\AMC@NCBcheck{6}
599 \newlength{\AMC@CBtaille}\setlength{\AMC@CBtaille}{5cm}
600 \def\AMC@premierecopie{1}
```

`\AMC@binaryCode` The command `\AMC@binaryCode{<options>}{<n>}` prints boxes to represent the number  $\langle n \rangle$  in its binary form. Options from `<options>` include:

`ndigits=<ndigits>` for the number of digits to be shown.

`id=<id>` for an ID of the number role (1 for the student number, 2 for the page number, 3 for the checking value).

`hsep=<hsep>` for the space between boxes.

`style=<style>` for some box style options.

`\AMCbin@one` and `\AMCbin@zero` print individual digit-boxes.

For example, `\AMC@binaryCode{ndigits=12}{367}` shows  $367 = 000101101111_2$  using 12 boxes:



```
601 \newtoks\AMCbin@sequence
602 \newcount\AMCbin@number
603 \newcount\AMCbin@digit
604 \newcount\AMCbin@id
605 \newcount\AMCbin@did
606 \newcount\AMCbin@ndigits
607 \newdimen\AMCbin@hsep
608 \define@key{AMCbin}{ndigits}{\AMCbin@ndigits=#1}
609 \define@key{AMCbin}{id}{\AMCbin@id=#1}
610 \define@key{AMCbin}{hsep}{\AMCbin@hsep=#1}
611 \define@key{AMCbin}{style}[]{\def\AMCbin@style{#1}}
612 \def\AMCbin@one{%
613   \ifnum\AMCbin@did>\z@%
614     \hspace{\AMCbin@hsep}%
615     \fi%
616   \advance\AMCbin@did\@ne%
617   \ifnum\AMCbin@id>0%
618     \AMC@answerBox@{}{\AMC@checkbox}{1}{chiffre:\the\AMCbin@id,\the\AMCbin@did}%
619     \else%
620     \AMC@answerBox@{}{\AMC@checkbox}{1}{}%
621     \fi}
622 \def\AMCbin@zero{%
623   \ifnum\AMCbin@did>\z@%
624     \hspace{\AMCbin@hsep}%
625     \fi%
626   \advance\AMCbin@did\@ne%
```

```

627 \ifnum\AMCbin@id>0%
628 \AMC@answerBox@{}{}{1}{chiffre:\the\AMCbin@id,\the\AMCbin@did}%
629 \else%
630 \AMC@answerBox@{}{}{1}{}%
631 \fi}
632 \newcommand{\AMC@binaryCode}[2]{%
633 \setkeys{AMCbin}{ndigits=1,hsep=0pt,style}\setkeys{AMCbin}{#1}%
634 \AMCbin@did=\z@%
635 {\AMCboxDimensions{shape=square,size=.32cm,down=0pt,rule=.2pt,cross=false}\expandafter\AMCboxDimensions\the\AMCbin@did}%
636 \AMCbin@digit=\z@%
637 \loop%
638 \ifnum\AMCbin@number>\z@%
639 \advance\AMCbin@digit\@ne%
640 \ifodd\AMCbin@number\AMCbin@sequence=\expandafter{\expandafter\AMCbin@one\the\AMCbin@sequence}%
641 \else\AMCbin@sequence=\expandafter{\expandafter\AMCbin@zero\the\AMCbin@sequence}\fi%
642 \divide\AMCbin@number\tw@%
643 \repeat%
644 \loop\relax%
645 \ifnum\AMCbin@digit<\AMCbin@endigits\advance\AMCbin@digit\@ne%
646 \AMCbin@sequence=\expandafter{\expandafter\AMCbin@zero\the\AMCbin@sequence}\repeat%
647 \the\AMCbin@sequence%
648 \ifnum\AMCbin@digit>\AMCbin@endigits\PackageError{automultiplechoice}{Too low AMC@NCB value (got \the\AMCbin@digit)}{}
649 }}

```

The commands `\AMCbin@begin` and `\AMC@binaryBoxes` are now unused and are defined for backward compatibility.

```

650 \def\AMCbin@begin#1{\setkeys{AMCbin}{id=#1}}
651 \newcommand{\AMC@binaryBoxes}[2][1]{%
652 \AMC@binaryCode{ndigits=#1}{#2}%
653 }

```

## 4.9 Checking Environment

`\AMCcurrentenv` Sets the current environment as document.

```

654 \def\AMCcurrentenv{document}

```

`\AMCif@env` Checks for the current environment.

```

655 \def\AMCif@env#1{
656 \def\AMC@tempenv{#1}%
657 \ifx\AMC@tempenv\AMCcurrentenv
658 \expandafter\@firstoftwo
659 \else
660 \expandafter\@secondoftwo
661 \fi
662 }

```

## 4.10 Handling groups of questions

The package allows to handle groups of questions, so as to be able to shuffle them before printing them to the sheets.

`\nouveau groupe` Command `\nouveau groupe{⟨group-name⟩}{⟨n⟩}` creates a new (empty) group with name `⟨group-name⟩` (argument `⟨n⟩` is present only for compatibility reasons and is ignored). Command `\element{⟨group-name⟩}{⟨text⟩}` adds to group `⟨group-name⟩` a new element that contains `⟨text⟩`. `⟨text⟩` can be a `question` environment, ore two successive `questions` to be kept together, or anything else. Calling command `\nouveau groupe` is not compulsory, as `\element` calls it if necessary.

```

663 \newcount\AMCtok@k
664 \newcount\AMCtok@max
665 \newcount\AMCtok@size
666 \newcommand{\nouveau groupe}[2]{%
667   \expandafter\ifx\csname #1@k\endcsname\relax%
668     \expandafter\newcount\csname #1@k\endcsname%
669     \expandafter\newcount\csname AMC#1@j\endcsname%
670     \csname #1@k\endcsname=\z@\relax%
671     \csname AMC#1@j\endcsname=\z@\relax%
672     \setgroupmode{#1}{\AMCdefault@groupmode}%
673   \fi%
674 }
675 \newcommand\AMC@prepare@element[1]{%
676   \nouveau groupe{#1}{}%
677   \global\advance\csname #1@k\endcsname\@ne\relax%
678   \AMCtok@k=\csname #1@k\endcsname%
679   \expandafter\ifx\csname #1@romannumeral\AMCtok@k\endcsname\relax%
680     \expandafter\newtoks\csname #1@romannumeral\AMCtok@k\endcsname\fi%
681 }
682 \newcommand{\element}[2]{%
683   \AMC@prepare@element{#1}%
684   \global\csname #1@romannumeral\AMCtok@k\endcsname={#2}%
685 }

```

`\setgroupmode` Command `\setgroupmode{⟨group-name⟩}{⟨mode⟩}` sets the group mode to `⟨mode⟩` for group `⟨group-name⟩`. This mode setup the behaviour of `\insertgroup` and `\copygroup` for this group:

1. With mode `fixed`, group's elements will be taken from the beginning.
2. With mode `cyclic`, the elements will be taken from the group following the last call group's use, recycling if necessary.
3. Mode `withreplacement` is the same as `fixed`, but the group is shuffled before each use.
4. Mode `withoutreplacement` is like `cyclic`, adding some shuffling when coming back to the beginning of the group.

The command `\setdefaultgroupmode{⟨mode⟩}` sets the group mode to be used for the following created groups (a group is created at the first `\element{⟨group⟩}` call). When no `\setdefaultgroupmode` is used, `fixed` is the default mode.

```

686 \def\AMCdefault@groupmode{fixed}
687 \newcommand{\setdefaultgroupmode}[1]{\def\AMCdefault@groupmode{#1}}
688 \newcommand{\setgroupmode}[2]{%
689   \expandafter\ifx\csname AMCgrouppre@#2\endcsname\relax%
690     \PackageError{automultiplechoice}{Unknown group mode for #1 : #2}%
691     {You asked to set group '#1' mode to '#2',
692      but '#2' is not a valid group mode}%

```

```

693 \else%
694 \expandafter\global\expandafter\def\csname AMC#1@mode\endcsname{#2}%
695 \fi%
696 }

```

The functions `\AMCgrouppre@xxx{<group-name>}{<n>}{<i>}` are called before using  $\langle n \rangle$  elements from group  $\langle group-name \rangle$  starting from index  $\langle i \rangle$  (negative value for  $\langle i \rangle$  stands for the current value of the group index), either with `\insertgroup` or `\copygroup`.

For mode **fixed**, the group index is set to  $\langle i \rangle$ , or 0 if  $\langle i \rangle$  is negative (take elements from the beginning).

```

697 \newcommand{\AMCgrouppre@fixed}[3]{%
698 \ifnum#3<\z@%
699 \csname AMC#1@j\endcsname=\z@%
700 \else%
701 \csname AMC#1@j\endcsname=#3%
702 \fi%
703 }

```

For mode **withreplacement**, the group is shuffled and the group index is set to  $\langle i \rangle$  or 0 (take elements from the beginning) if negative.

```

704 \newcommand{\AMCgrouppre@withreplacement}[3]{%
705 \ifnum#3<\z@%
706 \csname AMC#1@j\endcsname=\z@%
707 \else%
708 \csname AMC#1@j\endcsname=#3%
709 \fi%
710 \shufflegroup{#1}%
711 }

```

For mode **withoutreplacement**, the group index is set to  $\langle i \rangle$ , or left unchanged if  $\langle i \rangle$  is negative. If there is not enough elements left in the group, the elements before the index and the elements after the index are shuffled.

```

712 \newcount\AMC@imax
713 \newcommand{\AMCgrouppre@withoutreplacement}[3]{%
714 \ifnum#3<\z@%
715 \else%
716 \csname AMC#1@j\endcsname=#3%
717 \fi%
718 \ifnum\AMCtok@ik=\AMCloop@k%
719 \AMCtok@ik=\z@%
720 \fi%
721 \ifnum\AMCtok@ik=\z@%
722 \shufflegroup{#1}%
723 \else%
724 \AMC@imax=\AMCloop@k%
725 \advance\AMC@imax -#2\relax%
726 \ifnum\AMCtok@ik>\AMC@imax%
727 \shufflegroupslice{#1}{\@ne}{\AMCtok@ik}%
728 \ifnum\AMCtok@ik<\AMCloop@k%
729 \advance\AMCtok@ik\@ne%
730 \shufflegroupslice{#1}{\AMCtok@ik}{\AMCloop@k}%
731 \fi%
732 \fi%
733 \fi%

```

734 }

For mode **cyclic**, nothing has to be done, except setting the group index if non-negative.

735 \newcommand{\AMCgroup@pre@cyclic}[3]{%

736 \ifnum#3<\z%

737 \else%

738 \csname AMC#1@j\endcsname=#3%

739 \fi%

740 }

The function `\AMCgroup@pre{<mode>}{<group-name>}{<n>}{<i>}` calls the right `\AMCgroup@pre@xxx` command.

741 \newcommand{\AMCgroup@pre}[4]{%

742 \csname AMCgroup@pre@#1\endcsname{#2}{#3}{#4}%

743 }

`\shufflegroup` Command `\shufflegroup{<group-name>}` shuffles the elements of group `<group-name>`, and  
`\insertgroup` `\shufflegroupslice{<group-name>}{<a>}{<b>}` shuffles elements `<a>` to `<b>` from group `<group-name>`.  
`\insertgroupfrom` It can be called at each student sheet in order to get different student sheets and avoid cheating.

Command `\insertgroup[<n>]{<groupname>}` inserts all the elements of group `<groupname>`, or only the first `<n>` elements if `<n>` is given. `\insertgroupfrom[<n>]{<groupname>}{<i>}` inserts all the elements of group `<groupname>` starting from index `<i>` (the index of the first element is 0), or only the first `<n>` elements if `<n>` is given.

744 \newcommand{\shufflegroup}[1]{%

745 \ifAMC@shuffleG{\AMC@shuffletoks{number\csname #1@k\endcsname}{#1@}}\fi%

746 }

747 \newcommand{\shufflegroupslice}[3]{%

748 \ifAMC@shuffleG{\AMC@shuffletoks[#2]{#3}{#1@}}\fi%

749 }

750 \newcount\AMCtok@ik

751 \newcount\AMCloop@k

752 \newcommand{\AMCgroup@loop@prep}[3]{%

753 \AMCtok@size=#1\relax%

754 \ifAMC@fullGroups\AMCtok@size=\m@ne\fi%

755 \ifnum\AMCtok@size<\z%

756 \AMCtok@size=\csname #2@k\endcsname%

757 \fi%

758 \AMCtok@ik=\csname AMC#2@j\endcsname%

759 \AMCloop@k=\csname #2@k\endcsname%

760 \expandafter\ifx\csname AMC#2@mode\endcsname\relax%

761 \PackageError{automultiplechoice}{No group mode for #2}%

762 {No mode has been defined for group '#2'. This should not occur...}%

763 \fi%

764 \AMCgroup@pre{\csname AMC#2@mode\endcsname}{#2}{\the\AMCtok@size}{#3}%

765 }

766 \newcommand{\AMCgroup@loop@next}[1]{%

767 \global\advance\csname AMC#1@j\endcsname\@ne\relax%

768 \expandafter\ifnum\csname AMC#1@j\endcsname>\AMCloop@k\relax%

769 \global\csname AMC#1@j\endcsname=\@ne%

770 \fi%

771 \AMCtok@ik=\csname AMC#1@j\endcsname%

772 \advance\AMCtok@size\m@ne%

773 }

774 \newcommand{\insertgroupfrom}[3][-1]{%

```

775 \ifnum#1=0%
776 \else%
777 \AMCgrouploop@prep{#1}{#2}{#3}%
778 {\loop%
779 \AMCgrouploop@next{#2}%
780 {\the\csname #2@\romannumeral\AMCtok@ik\endcsname}%
781 \ifnum\AMCtok@size>\z@\repeat}%
782 \fi%
783 }
784 \newcommand{\insertgroup}[2][-1]{%
785 \insertgroupfrom[#1]{#2}{-1}%
786 }

```

`\cleargroup` The commands `\cleargroup` and `\copygroup` can also be used to make more complex questions combinations in the exams, allowing for example to ask the package to shuffle 3 questions taken at random from group `groupa` and 5 questions taken at random from group `groupb`.

`\cleargroup{<group>}` clears the group `<group>`, ereasing all of its elements.

`\copygroup[<n>]{<from>}{<to>}` copies `<n>` elements from group `<from>` to group `<to>`. If optional parameter `<n>` is not given, all the questions from group `<from>` are copied. `\copygroupfrom[<n>]{<from>}{<to>}{<i>}` copies `<n>` elements from group `<from>` to group `<to>`, starting from element at index `<i>` (the index of the first element is 0). If optional parameter `<n>` is not given, all the questions from group `<from>` are copied.

See section 3.4 for an illustration for these commands.

```

787 \newcommand{\cleargroup}[1]{%
788 \nouveau groupe{#1}{}%
789 \csname #1@k\endcsname=\z@\relax%
790 \csname AMC#1@j\endcsname=\z@\relax%
791 }
792 \newcommand{\copygroupfrom}[4][-1]{%
793 \ifnum#1=0%
794 \else%
795 \AMCgrouploop@prep{#1}{#2}{#4}%
796 {\loop%
797 \AMCgrouploop@next{#2}%
798 \AMC@prepare@element{#3}%
799 \global\csname #3@\romannumeral\AMCtok@k\endcsname=\csname #2@\romannumeral\AMCtok@ik\endcsname%
800 \ifnum\AMCtok@size>\z@\repeat}%
801 \fi%
802 }
803 \newcommand{\copygroup}[3][-1]{%
804 \copygroupfrom[#1]{#2}{#3}{-1}%
805 }

```

## 4.11 Questions

To manage multiple choice questions, first set some counters and token registers to handle answers. Token registers `\reponse@i`, `\reponse@ii` and so on will be used for answers – we restrict the number of answers of a single questions to `\AMCload@counter = 199`.

```

806 \newcount\AMCrep@count
807 \AMCload@counter=199
808 \@whilenum\AMCload@counter>0\do{%
809 \expandafter\newtoks\csname reponse@\romannumeral\AMCload@counter\endcsname%

```

```

810 \advance\AMCload@counter\m@ne%
811 }

```

`\AMCload@reponse` Command `\AMCload@reponse{<n>}{<text>}` will be used to add answer number `<n>` with text `<text>` (`<text>` will include the box to be ticked and all the layout commands) to the set of answers (in a token register `\reponse@xxx` – counter `\AMCload@counter` keeps track of the number of answers), in order to shuffle them when all answers will be loaded.

When answers are not to be shuffled, command `\AMCrien@deux{<n>}{<text>}` will be used instead, only printing `<text>`.

```

812 \newcommand\AMCload@reponse[2]{%
813   \global\advance\AMCload@counter\@ne\relax%
814   \global\csname reponse@\romannumeral\AMCload@counter\endcsname%
815   =\expandafter{\expandafter\AMCrep@count\expandafter=#2 #1}%
816 }
817 \newcommand\AMCrien@deux[2]{#1}

```

`\shuffle@it` After loading all answers, commands `\shuffle@it` will be used to shuffle them, and `\AMCdump@reponses` to print them.

```

818 \def\shuffle@it{\AMC@shuffletoks{\number\AMCload@counter}{reponse@}}
819 \newcount\AMCnum@questions
820 \newcommand\AMCdump@reponses{%
821   \global\AMCnum@questions=\AMCload@counter%
822   \@whilenum\AMCload@counter>0\do{%
823     \the\csname reponse@\romannumeral\AMCload@counter\endcsname%
824     \advance\AMCload@counter\m@ne}}

```

#### 4.11.1 Managing answers

`\lastchoices` Command `\AMCrep@init{<mode>}` is called for each question before reading answers. `<mode>` is `r` for suffled answers, and `o` if answers are not to be shuffled. It sets the number of answers counter to zero, and calls `\AMCrep@o` or `\AMCrep@r` depending on `<mode>`. These commands sets `\AMCload@@reponse` and `\AMCrep@fini` that will be called for each answer and after the last answer respectively, depending on `<mode>`:

- If `<mode>=r`, `\AMCload@@reponse` is `\AMCload@reponse` (loads answer to token register) and `\AMCrep@fini` calls `\shuffle@it` and `\AMCdump@reponses`;
- If `<mode>=o`, `\AMCload@@reponse` is `\AMCrien@deux` (prints answer directly) and `\AMCrep@fini` does nothing.

Command `\lastchoices` is called before giving answers that are to be printed at the end (even when shuffling answers). It closes the answers list calling `\AMCrep@fini` and opens another one in ordered mode. Note that it also saves the value of `\AMCrep@count`, which is the number of the current answer among all answers given in the subject source for the current question.

Command `\AMC@fin@rep` is to be called after the last answer: it adds a “None of these answers are correct.” answer if necessary (package option `completemulti`) with answer number zero, and calls `\AMCrep@fini`.

```

825 \newcommand\AMCrep@init[1]{%
826   \ifAMC@ordre\AMCrep@o\else%
827     \csname AMCrep@#1\endcsname\fi\AMCload@counter=\z@}
828 \newcommand\AMCrep@o{%
829   \def\AMCload@@reponse{\AMCrien@deux}\def\AMCrep@fini{}}

```

```

830 \newcommand\AMCrep@r{%
831   \def\AMCload@reponse{\AMCload@reponse}%
832   \def\AMCrep@fini{\shuffle@it\AMCdump@reponses}}
833 \newcount\AMCrep@count
834 \newcommand\lastchoices{%
835   \AMCrep@count=\AMCrep@count%
836   \AMCrep@fini\AMCrep@init{o}%
837   \AMCrep@count=\AMCrep@count}
838 \newcommand\@aucune{\emph{\AMC@loc@none}}
839 \newcommand\AMC@fin@rep{%
840   \ifAMCcomplete@multi\ifAMCtype@multi%
841     \lastchoices\AMCrep@count=-1%
842     \ifAMC@bonne\wrongchoice{\@aucune}\else%
843       \ifAMC@postcorrect\wrongchoice{\@aucune}\else\correctchoice{\@aucune}\fi%
844     \fi\fi\fi\AMCrep@fini}

```

#### 4.11.2 Separate answer sheet

This package needs some memory to print questions/answers boxes again on a separate answer sheet.

`\AMCformQuestion` First define commands that will announce questions and answers on the separate answer sheet (these commands can be modified by the user): `\AMCformQuestion{<number>}` is responsible for announcing question, and `\AMCformAnswer{<box>}` is responsible for printing the box to be ticked, given as argument `<box>`.

Commands `\AMCformQuestionA` and `\AMCformAnswerA` set up counter `\AMC@ncase` value before calling their counterparts.

```

845 \def\AMCformBeforeQuestion{\vspace{\AMCformVSpace}\par}
846 \def\AMCformAfterQuestion{\ifAMC@asqbloc\egroup\fi}
847 \def\AMCformQuestion#1{\AMC@loc@qf{#1}}
848 \def\AMCformQuestionN{\AMCformQuestion{\AMC@qaff}}
849 \def\AMCformQuestionA{%
850   \setcounter{AMC@ncase}{0}%
851   \AMCformBeforeQuestion%
852   \ifAMC@asqbloc\vbox\bgroup\fi%
853   \ifx\@empty\AMC@sza@callout\@empty\else%
854     \csname\AMC@sza@callout\endcsname%
855   \fi%
856   \AMCformQuestionN%
857   \ifx\@empty\AMC@sza@callin\@empty\else%
858     \csname\AMC@sza@callin\endcsname%
859   \fi%
860 }
861 \def\AMCformAnswer#1{\hspace{\AMCformHSpace} #1}
862 \def\AMCformAnswerA#1{\addtocounter{AMC@ncase}{1}\AMCformAnswer{#1}}

```

`\AMC@mem@add@ifneeded` These are commands to manage memory for separate answer sheet. `\AMC@mem@add@ifneeded{<code>}` adds `<code>` to this memory. `\AMC@mem@answer{<code>}` adds to memory answer code `<code>`, and `\AMCform` `\AMC@mem@openQuestion` adds to memory question code to announce current question.

`\AMCformS` The command `\AMCformBegin` defines the beginning of the separate answer sheet for the current student sheet, and `\AMCform` prints the whole memory: questions and answers boxes.



\AMCformS is a \AMCform variant that does not clear the list of answer boxes. It can be used to make the same exact subject for all students, displaying the questions before (outside) onecopy, so that onecopy contains only the answer sheet.

```

863 \ExplSyntaxOn
864
865 \prg_set_conditional:Nnn \amc_if_separate_question: { p , T } {
866   \ifAMC@ensemble
867     \ifAMC@zoneformulaire
868       \prg_return_false:
869     \else
870       \prg_return_true:
871     \fi
872   \else
873     \prg_return_false:
874   \fi
875 }
876 \cs_new_eq:NN \AMC@if@separate@question \amc_if_separate_question:T
877
878 \int_new:N \amc_memory_elts_count
879
880 \cs_new:Nn \amc_clear_memory: { \int_gzero:N \amc_memory_elts_count }
881 \cs_new_eq:NN \AMC@mem@clear \amc_clear_memory:
882
883 \cs_new:Npn \amc_memory_elt_i:n #1 {
884   amc_memory_elts_ \int_to_alph:n { #1 }
885 }
886 \cs_new:Nn \amc_memory_current_elt: {
887   \amc_memory_elt_i:n \amc_memory_elts_count
888 }
889 \cs_new:Npn \amc_memory_vars_i:n #1 {
890   amc_memory_vars_ \int_to_alph:n { #1 }
891 }
892 \cs_new:Nn \amc_memory_current_vars: {
893   \amc_memory_vars_i:n \amc_memory_elts_count
894 }
895
896 \cs_new:Nn \amc_add_memory_elt: {
897   \int_gincr:N \amc_memory_elts_count
898   \tl_gclear_new:c { \amc_memory_current_elt: }
899   \tl_gclear_new:c { \amc_memory_current_vars: }
900 }
901 \cs_new_eq:NN \AMC@mem@next \amc_add_memory_elt:
902
903 \cs_new:Npn \amc_add_to_memory:n #1 {
904   \tl_gput_right:cn { \amc_memory_current_elt: } { #1 }
905 }
906 \cs_new_eq:NN \AMC@mem@add \amc_add_to_memory:n
907
908 \cs_new:Npn \amc_add_to_vars:n #1 {
909   \tl_gput_right:cn { \amc_memory_current_vars: } { #1 }
910 }
911 \cs_new_eq:NN \AMC@mem@addvar \amc_add_to_vars:n
912

```

```

913 \cs_new:Npn \amc_add_qidaffname:nnn #1#2#3 {
914   \amc_add_to_vars:n { \AMCid@quest=#1\setcounter{AMCquestionaff}{#2}%
915     \global\def\AMCid@name{#3}}
916 }
917 \cs_generate_variant:Nn \amc_add_qidaffname:nnn { xxx }
918 \cs_new_eq:NN \AMC@mem@qidaffname \amc_add_qidaffname:xxx
919
920 \cs_new:Npn \amc_mem_elt_cat:n #1 {
921   \amc_add_to_vars:n { \def\AMCmem@elt@cat{ #1 } }
922 }
923 \cs_generate_variant:Nn \amc_mem_elt_cat:n { x }
924 \cs_new_eq:NN \AMC@mem@category \amc_mem_elt_cat:x
925
926 \cs_new:Npn \amc_add_aid:n #1 {
927   \amc_add_to_memory:n {\AMCrep@count=#1}
928 }
929 \cs_generate_variant:Nn \amc_add_aid:n { x }
930 \cs_new_eq:NN \AMC@mem@aid \amc_add_aid:x
931
932 \cs_new:Npn \amc_if_category_is_p:n #1 {
933   \str_if_eq_p:on { \AMCmem@elt@cat } { #1 }
934 }
935 \cs_new:Npn \amc_use_memory:n #1 {
936   \int_step_inline:nnnn { 1 } { 1 } \amc_memory_elts_count {
937     \def\AMCmem@elt@cat{ plain }
938     \tl_use:c { \amc_memory_vars_i:n { ##1 } }
939     \bool_if:nTF { #1 } {
940       \tl_use:c { \amc_memory_elt_i:n { ##1 } }
941     } { }
942   }
943 }
944 \cs_new:Nn \amc_use_memory: { \amc_use_memory:n { \c_true_bool } }
945 \cs_new_eq:NN \AMC@mem@show \amc_use_memory:
946 \cs_new_eq:NN \AMC@mem@show@filter \amc_use_memory:n
947 \cs_new_eq:NN \AMCifcategory \amc_if_category_is_p:n
948
949 \ExplSyntaxOff
950 \newcommand\AMC@mem@add@ifneeded[1]{%
951   \AMC@if@separate@question{%
952     \AMC@mem@add{#1}%
953   }%
954 }
955 \newcommand\AMC@mem@addsingle@ifneeded[2]{%
956   \AMC@if@separate@question{%
957     \AMC@mem@next%
958     \AMC@mem@category{#2}%
959     \AMC@mem@add{#1}%
960   }%
961 }
962 \newcommand\AMC@mem@answer[1]{%
963   \addtocounter{AMC@ncase}{1}%
964   \AMC@if@separate@question{%
965     \AMC@mem@aid{\the\AMCrep@count}%

```

```

966 \AMC@mem@add{\AMCformAnswerA{#1}}}%
967 }%
968 }
969 \newcommand\AMC@mem@openQuestion{%
970 \AMC@if@separate@question{%
971 \AMC@mem@next%
972 \AMC@mem@qidaffname{\the\AMCid@quest}{\arabic{AMCquestionaff}}{\AMCid@name}%
973 \AMC@mem@add{\AMCformQuestionA}%
974 }%
975 }
976 \def\AMCformBegin{%
977 \AMC@zoneformulairetrue\setcounter{section}{0}%
978 \ifAMC@ensemble\ifAMC@automarks\pagestyle{AMCpageFull}\fi\fi%
979 }
980 \newcommand\AMCform{%
981 \ifAMC@ensemble\AMCformulaire@dedanstrue%
982 \AMC@mem@show%
983 \fi}
984 \newcommand\AMCformFilter[1]{%
985 \ifAMC@ensemble\AMCformulaire@dedanstrue%
986 \AMC@mem@show@filter{#1}%
987 \fi}
988 \newif\ifAMC@keepmemory
989 \newcommand\AMCformS{%
990 \ifAMC@ensemble\AMCformulaire@dedanstrue%
991 \AMCmessage{BR=0}\AMC@mem@show%
992 \global\AMC@keepmemorytrue%
993 \fi}

```

`\AMCsection` The `\AMCsection` and `\AMCsubsection` commands issue their standard counterparts (`\section` and `\subsection` with the same argument, both in the subject *and* in the separate answer sheet.

```

994 \newcommand{\AMCsectionNumbered}[1]{%
995 \section{#1}\AMC@mem@addsingle@ifneeded{\section{#1}}{section}}
996 \newcommand{\AMCsubsectionNumbered}[1]{%
997 \subsection{#1}\AMC@mem@addsingle@ifneeded{\subsection{#1}}{subsection}}
998 \newcommand{\AMCsectionStar}[1]{%
999 \section*{#1}\AMC@mem@addsingle@ifneeded{\section*{#1}}{section}}
1000 \newcommand{\AMCsubsectionStar}[1]{%
1001 \subsection*{#1}\AMC@mem@addsingle@ifneeded{\subsection*{#1}}{subsection}}
1002 \def\AMCsection{\@ifstar\AMCsectionStar\AMCsectionNumbered}
1003 \def\AMCsubsection{\@ifstar\AMCsubsectionStar\AMCsubsectionNumbered}

```

### 4.11.3 Formatting answers

`choices` Answers have to be included in an environment `choices` (standard), `choiceshoriz` (answers on one line) or `choicescustom` (user defined) depending on the desired formatting.

`choicescustom` Use `\AMCBoxedAnswers` to request all answers to be included in  $\text{\LaTeX}$  boxes; this can be useful for example when using multicolumn answers formatting.

```

tikz-multi 1004 \def\AMCBoxedAnswers{\AMC@rbloctrue}
\AMCBoxedAnswers 1005 \newenvironment{choices}[1][r]{%
1006 \AMCrep@count=\z@ \def\une@rep{\AMCrep@itemize}%
1007 \ifAMC@rbloc \def\une@rep{\AMCrep@bloc}%
1008 \else \begin{itemize} \setlength{\itemsep}{\AMCinterIrep} \fi%

```

```

1009 \AMCrep@init{#1}}%
1010 {\AMC@fin@rep\ifAMC@rbloc\else\end{itemize}\fi}
1011 \newenvironment{choiceshoriz}[1][r]{%
1012 \AMCrep@count=\z@ \def\une@rep{\AMCrep@ligne}\AMCrep@init{#1}%
1013 \par\begin{center}}%
1014 {\AMC@fin@rep\end{center}}
1015 \newenvironment{choicescustom}[1][r]{%
1016 \AMCrep@count=\z@ \def\une@rep{\AMCrep@perso}\AMCrep@init{#1}%
1017 \AMCbeginAnswer\ignorespaces}%
1018 {\AMC@fin@rep\AMCendAnswer}
1019 \newenvironment{tikz-single}[1][r]{
1020 \AMCrep@count=\z@ \def\une@rep{\AMCrep@tikz}\AMCrep@init{#1}%
1021 \begin{tikzpicture}}{\AMC@fin@rep\end{tikzpicture}}
1022 \newenvironment{tikz-multi}[1][r]{
1023 \AMCrep@count=\z@ \def\une@rep{\AMCrep@tikzmat}\AMCrep@init{#1}%
1024 \begin{tikzpicture}[remember picture]}{\AMC@fin@rep\end{tikzpicture}}

```

`\AMCrep@bloc` For each of these styles, a corresponding `\AMCrep@xxx{⟨box⟩}{⟨text⟩}` is defined, which will  
`\AMCrep@tikz` format the answer with a box given in `⟨box⟩` and text `⟨text⟩`. `\AMCrep@bloc` is also defined and  
`\AMCrep@tikzmat` used in standard formatting when the user wants to put answers inside a L<sup>A</sup>T<sub>E</sub>X box.

```

\AMCrep@itemize 1025 \newcommand\AMCrep@bloc[2]{\AMC@mem@answer{#1}%
\AMCrep@ligne 1026 \par%
\AMCrep@perso 1027 \ifAMC@textPos\vbox\bgroup\AMC@tracepos{1}{atext:\AMCid@name:\the\AMCid@quest,\the\AMCrep@count}\hbox{b
1028 \noindent\begin{minipage}{\linewidth}%
1029 \begin{itemize}\item[#1] #2\end{itemize}\end{minipage}%
1030 \ifAMC@textPos\AMC@tracepos{1}{atext:\AMCid@name:\the\AMCid@quest,\the\AMCrep@count}\egroup\AMC@trace
1031 \vspace{\AMCinterBrep}}
1032 \newcommand\AMCrep@tikz[5]{\AMC@mem@answer{#1}\node[#4] (lab\thecsvrow) at (#3) {#2} node[#5] (box\thecsv
1033 \newcommand\AMCrep@tikzmat[5]{\AMC@mem@answer{#1}\node[#5] (box\thecsvrow) at (#3) {#1} node[#4] (lab\thec
1034 \newcommand\AMCrep@itemize[2]{\AMC@mem@answer{#1}\item[#1] #2}
1035 \newlength\AMChorizAnswerSep
1036 \setlength{\AMChorizAnswerSep}{3em plus 4em}
1037 \newlength\AMChorizBoxSep
1038 \setlength{\AMChorizBoxSep}{1em}
1039 \newcommand\AMCrep@ligne[2]{\AMC@mem@answer{#1}%
1040 \ifAMC@textPos%
1041 \mbox{\AMC@tracebox{1}{atext:\AMCid@name:\the\AMCid@quest,\the\AMCrep@count}{#1}\hspace*\AMChorizBoxSep}
1042 \else%
1043 \mbox{#1\hspace*\AMChorizBoxSep}#2}%
1044 \fi\hspace{\AMChorizAnswerSep}}
1045 \newcommand\AMCrep@perso[2]{\AMC@mem@answer{#1}\AMCanswer{#1}{#2}}

```

`\AMCbeginAnswer` The custom style will use user-defined commands to format answers: `\AMCbeginAnswer` is called  
`\AMCendAnswer` once before answers, `\AMCanswer{⟨box⟩}{⟨text⟩}` is called for each answer (`⟨box⟩` being the box  
`\AMCanswer` to be ticked and `⟨text⟩` the text associated with the proposed answer), and `\AMCendAnswer` is  
called after all answers.

```

1046 \def\AMCbeginAnswer{}
1047 \def\AMCanswer#1#2{#1 #2}
1048 \def\AMCendAnswer{}

```

`\answer` The commands `\correctchoice` and `\wrongchoice` are used inside `choices`-like environments  
`\correctchoice` to give the proposed answers and specify if they are to be ticked by the students or not.  
`\wrongchoice`

```

1049 \newcommand{\correctchoice}[2][\global\advance\AMCrep@count\@ne\relax%
1050   \ifAMC@calibration\AMCmessage{REP=\the\AMCrep@count:B}\fi%
1051   \global\AMC@bonnettrue%
1052   \AMCload@reponse{\une@rep{\ifAMC@correc\AMC@box{#1}\AMC@checkbox}%
1053     \else\AMC@box{#1}\fi}{#2}}{\the\AMCrep@count}\ignorespaces}
1054 \newcommand{\wrongchoice}[2][\global\advance\AMCrep@count\@ne\relax%
1055   \ifAMC@calibration\AMCmessage{REP=\the\AMCrep@count:M}\fi%
1056   \AMCload@reponse{\une@rep{\AMC@box{#1}\fi}{#2}}{\the\AMCrep@count}%
1057   \ignorespaces}

```

#### 4.11.4 Score zones

`\AMCscoreZone` The position of the scores on the annotated answer sheets can be defined in the L<sup>A</sup>T<sub>E</sub>X source file using `\AMCsetScoreZone{<options>}` (or `\AMCsetScoreZoneAnswerSheet{<options>}` for the answer sheets when the separate answer sheet option is used).

First begin with some helpers: `\AMCemptybox{<width>}{<height>}{<depth>}` draws an empty box with specified dimensions, and `\AMCmarginNote{<note>}` (code from one of [sgmoye's](http://tex.stackexchange.com) comments on [tex.stackexchange.com](http://tex.stackexchange.com)) prints a marginal note in the left or right margin, depending on current the position (usefull in `multicols` environment).

```

1058 \newcommand{\AMCemptybox}[3]{%
1059   \sbox0{\wd0=#1\ht0=#2\dp0=#3\relax\box0}}
1060 \newlength\AMC@mn@test
1061 \newlength\AMC@mn@sep\AMC@mn@sep=4mm
1062 \newlength\AMC@mn@leftmargin
1063 \newlength\AMC@mn@rightmargin
1064 \newcommand\AMCmarginNote[1]{%
1065   \begin{tikzpicture}[remember picture,overlay]%
1066     \coordinate (here) at (0,0);%
1067     \pgfextractx{\AMC@mn@test}{\pgfpointdiff{\pgfpointorigin}%
1068       {\pgfpointanchor{current page}{center}}}%
1069     \ifodd\thepage%
1070       \AMC@mn@leftmargin=\oddsidemargin%
1071       \AMC@mn@rightmargin=\evensidemargin%
1072     \else
1073       \AMC@mn@leftmargin=\evensidemargin%
1074       \AMC@mn@rightmargin=\oddsidemargin%
1075     \fi
1076     \ifdim\AMC@mn@test < 1cm%
1077       \draw (current page.east |- here)+(-\AMC@mn@rightmargin-1in+\AMC@mn@sep,0pt) node[anchor=text,align=right]{#1};
1078     \else%
1079       \draw (current page.west |- here)+(0cm,0pt) node[anchor=text,align=right,text width=\AMC@mn@leftmargin]{#1};
1080     \fi%
1081   \end{tikzpicture}%
1082 }

```

Define now different ways to place the score zone:

`none` nowhere

`question` right after the question heading

`margin` in the margin, using `marginpar` (this does not work with `multicols` environment)

`margins` in the left or right margin, depending on the current position (needs `tikz` package)

```

1083 \newcommand{\AMC@sz@box}{\AMCemptybox{\AMC@sz@width}{\AMC@sz@height}{\AMC@sz@depth}}
1084 %
1085 \newcommand{\AMC@sz@callin@question}{\AMCscoreZone{\AMC@sz@box}}
1086 %
1087 \newcommand{\AMC@sz@callout@margin}{\hspace{0pt}\marginpar{\AMCscoreZone{\AMC@sz@box}}}
1088 %
1089 \newcommand{\AMC@sz@init@margin}{\PackageWarning{automultiplechoice}{Please run twice to get proper marg
1090 \newcommand{\AMC@sz@callout@margin}{\hspace{0pt}\AMCmarginNote{\AMCscoreZone{\AMC@sz@box}}}

```

Let us now set up options handling.

```

1091 \newlength\AMC@sz@width
1092 \newlength\AMC@sz@height
1093 \newlength\AMC@sz@depth
1094 \def\AMC@sz@callout{}
1095 \def\AMC@sz@callin{}
1096 \define@key{AMC@sz}{width}{\AMC@sz@width=#1}
1097 \define@key{AMC@sz}{height}{\AMC@sz@height=#1}
1098 \define@key{AMC@sz}{depth}{\AMC@sz@depth=#1}
1099 \define@key{AMC@sz}{calloutside}{\def\AMC@sz@callout{#1}}
1100 \define@key{AMC@sz}{callinside}{\def\AMC@sz@callin{#1}}
1101 \define@choicekey{AMC@sz}{position}{none,question,margin,margins}{%
1102   \ifcsname AMC@sz@callout@#1\endcsname%
1103     \def\AMC@sz@callout{\AMC@sz@callout@#1}%
1104   \else%
1105     \def\AMC@sz@callout{}%
1106   \fi%
1107   \ifcsname AMC@sz@callin@#1\endcsname%
1108     \def\AMC@sz@callin{\AMC@sz@callin@#1}%
1109   \else%
1110     \def\AMC@sz@callin{}%
1111   \fi%
1112   \ifcsname AMC@sz@init@#1\endcsname%
1113     \csname AMC@sz@init@#1\endcsname%
1114   \fi%
1115 }
1116 \newcommand{\AMCsetScoreZone}[1]{\setkeys{AMC@sz}{#1}}
1117 \AMCsetScoreZone{width=1.5em,height=1.5ex,depth=.5ex,position=none}

```

And do the same for \AMCsetScoreZoneAnswerSheet...

```

1118 \newcommand{\AMC@sza@box}{\AMCemptybox{\AMC@sza@width}{\AMC@sza@height}{\AMC@sza@depth}}
1119 %
1120 \newcommand{\AMC@sza@init@none}{}
1121 \newcommand{\AMC@sza@callout@none}{}
1122 \newcommand{\AMC@sza@callin@none}{}
1123 %
1124 \newcommand{\AMC@sza@init@question}{}
1125 \newcommand{\AMC@sza@callout@question}{}
1126 \newcommand{\AMC@sza@callin@question}{\AMCscoreZone{\AMC@sza@box}}
1127 %
1128 \newcommand{\AMC@sza@init@margin}{}
1129 \newcommand{\AMC@sza@callout@margin}{\hspace{0pt}\marginpar{\AMCscoreZone{\AMC@sza@box}}}
1130 \newcommand{\AMC@sza@callin@margin}{}
1131 %
1132 \newcommand{\AMC@sza@init@margins}{\PackageWarning{automultiplechoice}{Please run twice to get proper mar

```

```

1133 \newcommand{\AMC@sza@callout@margin}{\hspace{0pt}\AMCmarginNote{\AMCscoreZone{\AMC@sz@box}}}
1134 \newcommand{\AMC@sza@callin@margin}{\hspace{0pt}\AMCmarginNote{\AMCscoreZone{\AMC@sz@box}}}
1135 %
1136 \newlength\AMC@sza@width
1137 \newlength\AMC@sza@height
1138 \newlength\AMC@sza@depth
1139 \def\AMC@sza@callout{}
1140 \def\AMC@sza@callin{}
1141 \define@key{AMC@sza}{width}{\AMC@sza@width=#1}
1142 \define@key{AMC@sza}{height}{\AMC@sza@height=#1}
1143 \define@key{AMC@sza}{depth}{\AMC@sza@depth=#1}
1144 \define@key{AMC@sza}{calloutside}{\def\AMC@sza@callout{#1}}
1145 \define@key{AMC@sza}{callinside}{\def\AMC@sza@callin{#1}}
1146 \define@choicekey{AMC@sza}{position}{none,question,margin,margins}{%
1147   \ifcsname AMC@sza@callout@#1\endcsname%
1148     \def\AMC@sza@callout{AMC@sza@callout@#1}%
1149   \else%
1150     \def\AMC@sza@callout{}%
1151   \fi%
1152   \ifcsname AMC@sza@callin@#1\endcsname%
1153     \def\AMC@sza@callin{AMC@sza@callin@#1}%
1154   \else%
1155     \def\AMC@sza@callin{}%
1156   \fi%
1157   \ifcsname AMC@sza@init@#1\endcsname%
1158     \csname AMC@sza@init@#1\endcsname%
1159   \fi%
1160 }
1161 \newcommand{\AMCsetScoreZoneAnswerSheet}[1]{\setkeys{AMC@sza}{#1}}
1162 \AMCsetScoreZoneAnswerSheet{width=1.5em,height=1.5ex,depth=.5ex,position=none}
1163 \newcommand{\AMCnoScoreZone}{\AMCsetScoreZone{position=none}\AMCsetScoreZoneAnswerSheet{position=none}}

```

#### 4.11.5 Formatting questions

`\AMCquestionaff` The counter `\AMCquestionaff` keeps track of the current question number. It can be redefined by the user, for example to print several questions without a number, and then print questions with a number starting at one.

`\AMC@stepQuestion` will increase this counter and `\AMC@qaff` will format the question number out.

```

1164 \newcounter{AMCquestionaff}
1165 \newcommand{\AMCnumero}[1]{\setcounter{AMCquestionaff}{#1}\addtocounter{AMCquestionaff}{-1}}
1166 \AtBeginDocument{%
1167   \ifx\@skiphyperreftrue\@undefined%
1168     \expandafter\newif\csname if@skiphyperref\endcsname%
1169   \fi%
1170 }
1171 \newcommand\AMC@stepQuestion{\ifAMCquestionNumber\@skiphyperreftrue\refstepcounter{AMCquestionaff}\@skiphyperreftrue}
1172 \newcommand\AMC@qaff{\arabic{AMCquestionaff}}

```

`\AMCbeforeQuestion` The command `\AMCbeforeQuestion` opens a new question. The command `\AMCbeginQuestion{<n>}{<sign>}` will format the question header, where `<n>` is the question number and `<sign>` being `\multiSymbole` in case of a multiple question, and empty in case of a simple one. `\AMCbeforeQuestion`, `\AMCbeginQuestion` and `\multiSymbole` can be user-redefined.

```

1173 \def\AMCbeforeQuestion{\ifAMC@qbloc\else\par\noindent\fi}
1174 \def\AMCbeginQuestion#1#2{\noindent\AMC@loc@q{#1}{#2}%
1175   \ifx\@empty\AMC@sz@callin\@empty\hspace*{1em}\fi%
1176 }
1177 \def\multiSymbole{$\clubsuit$}

```

**question** Environment `{question}{<key>}` encloses a simple question (with one and only one correct choice) with associated unique key `<key>` and the proposed answers.

**questionmult** Environment `{questionmult}{<key>}` is the same for multiple questions (with none, one or several correct choices).

**variable-single** Environment `{questionmultx}{<key>}` is the same as `questionmult`, but with no use of `\multiSymbole`.

**variable-multi** Environment `{questionmultx}{<key>}` is the same as `questionmult`, but with no use of `\multiSymbole`.

**questionouverte** Environment `{questionouverte}[<width>]` is used for open questions (that won't be marked automatically!), with width given as an optional argument (defaults to 3 cm).

**\ouverte@vs** Environment `{questionouverte}[<width>]` is used for open questions (that won't be marked automatically!), with width given as an optional argument (defaults to 3 cm).

The command `\AMCexternalQuestion{<id>}{<maxscore>}` allows to declare a question that will be scored outside AMC, with a maximal score `<maxscore>`. When you use this command, you can manage the question number and question text freely (AMC won't handle this).

```

1178 \ifx\question\undefined\else\let\question\undefined\fi
1179 \def\AMCnobloc{\AMC@qblocfalse}
1180 \def\AMCbloc{\AMC@qbloctrue}
1181 \newcommand\AMCstartWithQuestion[1]{%
1182   \global\def\AMCid@name{#1}\AMC@affecte{#1}{\AMCid@quest}%
1183   \ifAMC@calibration%
1184     \AMCmessage{Q=\the\AMCid@quest}%
1185     \immediate\write\AMC@XYFILE{\string\question{\the\AMCid@quest}{\AMCid@name}}%
1186   \fi%
1187 }
1188 \newcommand\AMCexternalQuestion[2]{%
1189   \AMCstartWithQuestion{#1}%
1190   \ifAMC@calibration%
1191     \AMCmessage{B=MAX=#2}%
1192     \AMCmessage{MULT}%
1193     \AMCmessage{FQ}%
1194   \fi%
1195 }
1196 \newenvironment{question}[2][{}]{%
1197   \def\AMCcurrentenv{question}%
1198   \AMC@stepQuestion%
1199   \AMCstartWithQuestion{#2}%
1200   \AMCbeforeQuestion%
1201   \ifx\@empty\AMC@sz@callout\@empty\else%
1202     \csname\AMC@sz@callout\endcsname%
1203   \fi%
1204   \AMCtype@multifalse\ifAMC@qbloc\ifAMC@textPos\vbox\bgroup\AMC@tracapos{1}{qtext:#2:\the\AMCid@quest,0}\
1205   \ifAMC@affichekeys\index{\texttt{#2}}\ifAMC@keyline[\texttt{#2}]\newline\fi\fi%
1206   \AMCbeginQuestion{\ifAMC@affichekeys\ifAMC@ensemble\AMC@qaff\ \fi\ifAMC@keyline\else[\texttt{#2}]\fi\e
1207   \ifx\@empty\AMC@sz@callin\@empty\else%
1208     \csname\AMC@sz@callin\endcsname%
1209   \fi%
1210   \AMCformulaire@dedansfalse\setcounter{AMC@ncase}{0}%
1211   \AMC@mem@openQuestion}%
1212 {\ifAMC@qbloc\end{minipage}\ifAMC@textPos\AMC@tracapos{1}{qtext:\AMCid@name:\the\AMCid@quest,0}\egroup\AM
1213 \newenvironment{questionmult}[1]{%

```



```

1214 \AMCune@bonnefalse\begin{question}[{\multiSymbole}]{#1}%
1215 \AMCtype@multitrue\ifAMC@calibration%
1216 \AMCmessage{MULT}\fi}%
1217 {\end{question}}
1218 \newenvironment{variable-single}[2]
1219 {\def\AMCbeginQuestion##1##2{}
1220 \begin{questionmult}{#1}\scoring{v=#2}
1221 \begin{tikz-single}[o]}
1222 {\end{tikz-single}
1223 \end{questionmult}}
1224 \newenvironment{variable-multi}[4]
1225 {\def\AMCbeginQuestion##1##2{}
1226 \begin{questionmult}{#1}\scoring{v=#4}
1227 \begin{tikz-multi}[o]
1228 \node[#3] (var) at (0,0) {#2};}
1229 {\end{tikz-multi}
1230 \end{questionmult}}
1231 \newenvironment{questionmultx}[1]{%
1232 \begingroup\def\multiSymbole{\begin{questionmult}{#1}}%
1233 {\end{questionmult}\endgroup}
1234 \newdimen\ouverte@vs
1235 \newenvironment{questionouverte}[1][3cm]{%
1236 \AMC@stepQuestion%
1237 \AMCtype@multifalse\ouverte@vs=#1%
1238 \ifAMC@qbloc\noindent\begin{minipage}{\linewidth}\fi%
1239 \AMCbeginQuestion{\AMC@qaff}{}}%
1240 {\vspace*{\ouverte@vs}\ifAMC@qbloc\end{minipage}\vspace{3ex}\fi}

```

#### 4.11.6 Explanations

**\explain** The command `\explain{<text>}` is used inside question-like environments to give the explanation for the answers of a question. The command `\explaincontext{<text>}` inserts its argument only in the corrected paper.

```

1241 \newcommand{\explain}[1]{%
1242 \ifAMC@correchead%
1243 \AMCif@env{question}{\par\noindent{\AMC@loc@explain #1}}{\AMC@error@explain}\vspace{1ex}%
1244 \else%
1245 \AMCif@env{question}{}{\AMC@error@explain}%
1246 \fi%
1247 }
1248 \newcommand{\explaincontext}[1]{%
1249 \ifAMC@correc%
1250 #1%
1251 \fi%
1252 }

```

#### 4.12 Scoring

**\scoring** Scoring strategies are simply transmitted to the .amc file for later analysis.

**\scoringDefaultS** `\scoring{<score>}` details the scoring strategy for current question or current answer,  
**\scoringDefaultM** `\scoringDefaultS{<score>}` and `\scoringDefaultM{<score>}` gives default scoring strategy for  
**uestionIndicative** simple and multiple questions, and `\QuestionIndicative` tells that the current question is not

no be taken into account in the global mark.

```
1253 \def\scoring#1{\ifAMC@calibration\AMCmessage{B=#1}\fi}
1254 \def\scoringDefaultS#1{\ifAMC@calibration\AMCmessage{BDS=#1}\fi}
1255 \def\scoringDefaultM#1{\ifAMC@calibration\AMCmessage{BDM=#1}\fi}
1256 \def\QuestionIndicative{\ifAMC@calibration\AMCmessage{INDIC}\fi}
```

## 4.13 Numerical data

### 4.13.1 Codes

`\AMCcodeGrid` Students can code some numerical information (such as student number) through special questions, which can be formatted easily with the command `\AMCcodeGrid[<opts>]{<key>}{<descr>}`, where *<key>* is a key prefix and *<descr>* is a coma-separated list of character pools to offer. The characters entered by the student will be available through the questions *<key>*[1], ..., *<key>*[*<length(descr)>*].

As an example,

`\AMCcodeGrid{code}{ABCD,012345,012345,012345,012345}` produces the opposite boxes (two results are show here: without or with `separateanswersheet` option), and trace positions of all the boxes in the .xy file with the `code` identifier: the first digit is represented by question with key `code[6]`, the second by question with key `code[5]`, and so on.

Positions of the boxes are logged in the .xy file, as shown in section 5.3 for the first set of boxes (without `separateanswersheet`, with digits outside boxes).

	0		0		0		0
	1		1		1		1
A	2		2		2		2
B	3		3		3		3
C	4		4		4		4
D	5		5		5		5

	0		0		0		0
	1		1		1		1
A	2		2		2		2
B	3		3		3		3
C	4		4		4		4
D	5		5		5		5

The “horizontal” version can also be considered using option `h`, especially with a small number of digits. See opposite for the result of

`\AMCcodeGrid[h]{code}{ABCDEF,0123456789,0123456789}`.

The `\AMCcodeGridInt[<opts>]{<key>}{<n>}` is a shortcut for calling `\AMCcodeGrid` with *<n>* digits from 0 to 9. This allows to create grids for *<n>*-digits integers easily.

These two commands supports the following options (given as a comma-separated list optional argument *<opts>*):

- `vertical=true` or `false` to indicate the direction to be used (default is `true`);
- `h` is a shortcut for `vertical=false`;
- `v` is a shortcut for `vertical=true`;
- `top` to request top-aligned columns in vertical direction.
- `multi` for codes that are repeated on each page.

```
1257 \newcount\AMC@chiffres
1258 \newdimen\AMCcodeHspace\AMCcodeHspace=.5em
```

```

1259 \newdimen\AMCcodeVspace\AMCcodeVspace=.5em
1260 \newcommand\AMCcodeID@squarebrackets[2]{#1[#2]}
1261 \newcommand\AMCcodeID@dot[2]{#1.#2}
1262 \newcommand\AMCcodeID@[1]{%
1263   \expandafter\def\expandafter\AMCcodeID\expandafter{\csname AMCcodeID@#1\endcsname}%
1264   \def\AMCcodeID@mode{#1}%
1265 }
1266 \AMCcodeID@[squarebrackets]
1267 \ExplSyntaxOn
1268
1269 \clist_new:N \amc_code_descr_clist
1270 \seq_new:N \amc_code_digits_seq
1271 \int_new:N \amc_code_digit_n_int
1272 \bool_new:N \amc_code_vertical_bool
1273 \bool_new:N \amc_code_top_bool
1274 \bool_new:N \amc_code_multi_bool
1275 \clist_new:N \amc__multi_clist
1276
1277 \cs_new:Npn \amc_code_init:N #1 {
1278   \def\AMCbeginQuestion##1##2{}
1279   \def\AMCbeforeQuestion{}
1280   \AMCnoScoreZone
1281   \AMCquestionNumberfalse
1282   \setlength{\parindent}{0pt}
1283   \AMCnobloc
1284   \int_set:Nn \amc_code_digit_n_int { \clist_count:N #1 }
1285 }
1286
1287 \cs_new:Nn \amc_code_digit_init: {
1288   \QuestionIndicative
1289   \global\AMCrep@count=\z@
1290 }
1291
1292 \cs_new:Npn \amc_code_digit:n #1 {
1293   \global\advance\AMCrep@count\@ne\relax
1294   \ifAMC@calibration\AMCmessage{ REP = \the\AMCrep@count : M }\fi
1295   \hbox{\AMC@keyBox@{#1}{1}{case : \AMCid@name : \the\AMCid@quest , \the\AMCrep@count}}
1296   \bool_if:NTF \amc_code_vertical_bool {
1297     \vspace{\AMCcodeVspace}
1298   }{
1299     \hspace{\AMCcodeHspace}
1300   }
1301 }
1302
1303 \keys_define:nn { amccode } {
1304   vertical .bool_set:N = \amc_code_vertical_bool,
1305   vertical .initial:n = { true },
1306   vertical .default:n = { true },
1307   v .code:n = { \bool_set_true:N \amc_code_vertical_bool },
1308   h .code:n = { \bool_set_false:N \amc_code_vertical_bool },
1309   top .bool_set:N = \amc_code_top_bool,
1310   top .initial:n = { false },
1311   top .default:n = { true },

```

```

1312 multi .bool_set:N = \amc_code_multi_bool,
1313 multi .initial:n = { false },
1314 multi .default:n = { true }
1315 }
1316
1317 \cs_new_nopar:Nn \amc_multi_report: {
1318   \ifAMC@calibration
1319     \immediate\write\AMC@XYFILE{\string\with{multi=\clist_use:Nn\amc__multi_clist{,}}}}
1320   \fi
1321 }
1322 \cs_new_eq:NN \AMC@multi@report \amc_multi_report:
1323 \int_new:N \amc_multi_count_int
1324 \cs_new_nopar:Nn \amc_multi_clear: {
1325   \int_gzero:N \amc_multi_count_int
1326 }
1327 \cs_new_eq:NN \AMC@multiclear \amc_multi_clear:
1328
1329 \cs_new:Npn \amc_code_generate:nNn #1#2#3 {
1330   { \keys_set:nn { amccode } { #3 }
1331     \bool_if:NTF \amc_code_multi_bool {
1332       \clist_gset:Nn \amc__multi_clist { #1 }
1333     } {}
1334     \bool_if:NTF \amc_code_multi_bool { \int_gincr:N \amc_multi_count_int } {}
1335     \amc_code_init:N #2
1336     \clist_map_inline:Nn #2 { % iterates over 'digits'
1337       \begin{question}{
1338         \AMCcodeID{ #1 \bool_if:NTF
1339           \amc_code_multi_bool
1340           { * \int_use:N \amc_multi_count_int } {} }
1341         { \int_use:N \amc_code_digit_n_int }
1342       }
1343       \amc_code_digit_init:
1344       \seq_set_split:Nnn \amc_code_digits_seq {} { ##1 }
1345       \bool_if:NTF \amc_code_vertical_bool {
1346         \hspace{0pt}
1347         \bool_if:NTF \amc_code_top_bool { \vtop } { \vbox }
1348         \bgroup
1349       }{
1350         \hbox\bgroup
1351       }
1352       \seq_map_inline:Nn \amc_code_digits_seq {
1353         % iterates over available characters for 'digit'
1354         \amc_code_digit:n { #####1 }
1355       }
1356       \bool_if:NTF \amc_code_vertical_bool {
1357         \vspace{-\AMCcodeVspace}\egroup
1358         \hspace{\AMCcodeHspace}
1359       }{
1360         \egroup\vspace{\AMCcodeVspace}
1361         \par
1362       }
1363     \end{question}
1364     \int_decr:N \amc_code_digit_n_int

```

```

1365     }
1366   }
1367 }
1368
1369 \cs_new:Npn \amc_code_generate:nnn #1#2#3 {
1370   \clist_set:Nn \amc_code_descr_clist { #2 }
1371   \amc_code_generate:nNn { #1 } \amc_code_descr_clist { #3 }
1372 }
1373 \cs_generate_variant:Nn \amc_code_generate:nnn { xxx }
1374 \newcommand{\AMCcodeGrid}[3][{}]{
1375   \amc_code_generate:xxx { #2 } { #3 } { #1 }
1376 }
1377
1378 \cs_new:Npn \amc_code_generate_integer:nnn #1#2#3 {
1379   \clist_clear:N \amc_code_descr_clist
1380   \prg_replicate:nn { #2 } { \clist_put_right:Nn \amc_code_descr_clist { 0123456789 } }
1381   \amc_code_generate:nNn { #1 } \amc_code_descr_clist { #3 }
1382 }
1383 \cs_generate_variant:Nn \amc_code_generate_integer:nnn { xxx }
1384 \newcommand{\AMCcodeGridInt}[3][{}]{
1385   \amc_code_generate_integer:xxx { #2 } { #3 } { #1 }
1386 }
1387
1388 \cs_new:Npn \amc_code_generate_integer_v:nn #1#2 {
1389   \amc_code_generate_integer:nnn { #1 } { #2 } { v }
1390 }
1391 \cs_new:Npn \amc_code_generate_integer_h:nn #1#2 {
1392   \amc_code_generate_integer:nnn { #1 } { #2 } { h }
1393 }
1394 \cs_generate_variant:Nn \amc_code_generate_integer_v:nn { xx }
1395 \cs_generate_variant:Nn \amc_code_generate_integer_h:nn { xx }
1396 \cs_new_eq:NN \AMCcode \amc_code_generate_integer_v:xx
1397 \cs_new_eq:NN \AMCcodeH \amc_code_generate_integer_h:xx
1398
1399 \ExplSyntaxOff

```

#### 4.13.2 Numerical questions

**\AMCnumericChoices** The command `\AMCnumericChoices{<correct>}{<options>}` can be used as a replacement for the `choices` environment when the questions asks for a numeric value to code on the answer sheet.

As an example,

```

\begin{question}{product}
  What is the value of  $7 \times 5$ ?
  \AMCnumericChoices{35}{digits=2,sign=false}
\end{question}

```

produces (in correction mode):

**Question 11** What is the value of  $7 \times 5$ ?

<input type="text"/>	0	<input type="text"/>	1	<input type="text"/>	2	<input checked="" type="text"/>	3	<input type="text"/>	4	<input type="text"/>	5	<input type="text"/>	6	<input type="text"/>	7	<input type="text"/>	8	<input type="text"/>	9
<input type="text"/>	0	<input type="text"/>	1	<input type="text"/>	2	<input type="text"/>	3	<input type="text"/>	4	<input checked="" type="text"/>	5	<input type="text"/>	6	<input type="text"/>	7	<input type="text"/>	8	<input type="text"/>	9

and these boxes are only shown on the separate answer sheet if the `separateanswersheet` option is used.

This command uses the `\AMCformatChoices{<showcommand>}{<hidecommand>}{<arg1>}{<arg2>}` command, that calls either `<hidecommand>{<arg1>}{<arg2>}` if the `separateanswersheet` option is used and if we are currently in the question part (not in the answer sheet), or `<showcommand>{<arg1>}{<arg2>}` when all the boxes are to be produced.

```

1400 \newcommand\AMCformatChoices[4]{%
1401   \global\AMCrep@count=\z@%
1402   \AMC@if@separate@question{%
1403     \AMC@mem@add{\global\AMCrep@count=\z@%
1404       #1{#3}{#4}}%
1405   }%
1406   \ifAMC@ensemble%
1407     #2{#3}{#4}%
1408     \AMCmessage{QPART}%
1409   \else%
1410     #1{#3}{#4}%
1411   \fi%
1412 }
```

Some computation commands are now defined. The command `\amc_fp_decompose:NNn{<fp var>}{<int var>}{<x>}` sets `<fp var>` to be the *mantissa* and `<int var>` the *exponent* of the floating point number `<x>`. For example, `\amc_fp_decompose:NNn\mant_fp\expo_int{123.456}` give the value 1.23456 to `\mant_fp` and 2 to `\expo_int` (because  $123.456 = 1.23456 \times 10^2$ ).

The command `\amc_fp_to_digits:Nnn{<clist>}{<x>}{<n digits>}{<base>}` rounds the floating point number `<x>` and populates the comma separated list `<clist>` with its `<n digits>` digits in base `<base>`. An error is issued if `<x>` would have required more digits.

```

1413 \ExplSyntaxOn
1414
1415 \cs_generate_variant:Nn \tl_replace_once:Nnn { Nxn }
1416
1417 \tl_new:N \amc_ee_tl
1418 \seq_new:N \amc_ee_seq
```

Note that with some versions of `l3fp-convert` (prior to 2017-09-18), `\fp_to_scientific` leads to a ‘e’ with catcode 12 (*other*). We convert it to catcode *letter* before splitting.

```

1419 \group_begin:
1420 \char_set_catcode_other:N E
1421 \tex_lowercase:D
1422 {
1423   \cs_new:Npn \amc_read_scientific:NNn #1 #2 #3 {
1424     \tl_set:Nn \amc_ee_tl { #3 }
1425     \tl_replace_once:Nxn \amc_ee_tl { E } { e }
1426     \seq_set_split:NnV \amc_ee_seq e \amc_ee_tl
1427     \fp_set:Nn #1 { \seq_item:Nn \amc_ee_seq 1 }
1428     \int_set:Nn #2 { \seq_item:Nn \amc_ee_seq 2 }
```

```

1429 }
1430 }
1431 \group_end:
1432
1433 \cs_generate_variant:Nn \amc_read_scientific:NNn { NNf, NNx }
1434
1435 \fp_new:N \amc_fulls_fp
1436 \cs_new:Npn \amc_fp_decompose:NNn #1 #2 #3 {
1437   \fp_set:Nn \amc_fulls_fp { #3 }

```

Note that with some versions of 13fp-convert, the exponent part is omitted for some values, so that we add e 0.

```

1438   \amc_read_scientific:NNx #1 #2
1439   { \fp_to_scientific:N \amc_fulls_fp e 0 }
1440 }
1441 \cs_generate_variant:Nn \amc_fp_decompose:NNn { NNx }
1442
1443 \fp_new:N \amc_num_mantissa_fp
1444 \int_new:N \amc_num_exponent_int
1445 \cs_new:Npn \amc_fp_n_significant_digits:Nnn #1 #2 #3 {
1446   \amc_fp_decompose:NNn \amc_num_mantissa_fp \amc_num_exponent_int
1447   { #2 }
1448   \fp_set:Nn #1
1449   { round(\amc_num_mantissa_fp * 10^((#3)-1)) }
1450   \fp_compare:nTF { abs(#1) >= 10^(#3) }
1451   {
1452     \fp_set:Nn #1 { #1 / 10 }
1453   } { }
1454 }
1455
1456 \fp_new:N \amc_num_nsig_fp
1457 \cs_new:Npn \amc_fp_show_n_significant_digits:nn #1 #2 {
1458   \amc_fp_n_significant_digits:Nnn \amc_num_nsig_fp { #1 } { #2 }
1459 }
1460 \cs_new_eq:NN \AMCsignificantDigits \amc_fp_show_n_significant_digits:nn
1461
1462 \cs_new:Npn \amc_fp_show_significant_digits: {
1463   \fp_use:N \amc_num_nsig_fp
1464 }
1465 \cs_new_eq:NN \AMCshowSignificantDigits \amc_fp_show_significant_digits:
1466
1467 \cs_new:Npn \amc_fp_n_digits:Nnn #1 #2 #3 {
1468   \fp_set:Nn #1
1469   { round((#2) * 10^(#3)) }
1470 }
1471
1472 \int_new:N \amc_todigits_int
1473 \cs_new:Npn \amc_fp_to_digits:Nnnn #1 #2 #3 #4 {
1474   \clist_clear:N #1
1475   \int_set:Nn \amc_todigits_int { \fp_eval:n { abs(round(#2)) } }
1476   \prg_replicate:nn { #3 } {
1477     \clist_put_left:Nx #1 { \int_mod:nn \amc_todigits_int { #4 } }
1478     \int_set:Nn \amc_todigits_int
1479     { \int_div_truncate:nn \amc_todigits_int { #4 } }

```

```

1480 }
1481 \int_compare:nNnTF \amc_todigits_int = 0 { } {
1482   \message{^^J!~Error:~number~too~large,
1483     ~some~digits~will~be~discarded^^J}
1484 }
1485 }
1486
1487 \cs_new:Npn \amc_invalid_digits:Nn #1 #2 {
1488   \clist_clear:N #1
1489   \prg_replicate:nn { #2 } { \clist_put_left:Nx #1 { -1 } }
1490 }
1491
1492 \cs_new:Npn \amc_get_fp_sign:Nn #1 #2 {
1493   \fp_compare:nNnTF #2 < 0 {
1494     \int_set:Nn #1 { -1 }
1495   }{
1496     \fp_compare:nNnTF #2 > 0 {
1497       \int_set:Nn #1 { 1 }
1498     }{
1499       \int_set:Nn #1 { 0 }
1500     }
1501   }
1502 }
1503
1504 \cs_new:Npn \amc_get_int_sign:Nn #1 #2 {
1505   \int_compare:nNnTF #2 < 0 {
1506     \int_set:Nn #1 { -1 }
1507   }{
1508     \int_compare:nNnTF #2 > 0 {
1509       \int_set:Nn #1 { 1 }
1510     }{
1511       \int_set:Nn #1 { 0 }
1512     }
1513   }
1514 }
1515
1516 \ExplSyntaxOff

```

The command `\AMCnumericShow{<value>}{<opts>}` is called to draw all necessary boxes to code a numerical value `<value>` with options given as a comma separated list `<opts>`. `\AMCnumericOpts{<opts>}` can be used to set some default values for these options.

Begin with the available options:

```

1517 \def\AMCncontextGoto{}
1518 \def\AMCncontextVHead#1{\emph{b#1}}
1519 \newdimen\AMCnumeric@Hspace\AMCnumeric@Hspace=.5em
1520 \newdimen\AMCnumeric@Vspace\AMCnumeric@Vspace=1ex
1521 \ExplSyntaxOn
1522
1523 \keys_define:nn { amcnumeric } {
1524   Tsign .code:n = {\def\AMCncontextSign{#1}},
1525   Tsign .initial:n = {},
1526   Tpoint .code:n = {\def\AMCdecimalPoint{#1}},
1527   Tpoint .initial:n = { \raisebox{1ex}{\bf .} },
1528   Texponent .code:n = {\def\AMCexponent{#1}},

```



```

1529 Texponent .initial:n = {  $\times 10$  \textasciicircum },
1530 vspace .code:n = {\AMCnumeric@Vspace=#1},
1531 hspace .code:n = {\AMCnumeric@Hspace=#1},
1532 bordercol .code:n = {\def\AMCncol@Border{#1}},
1533 bordercol .initial:n = { lightgray },
1534 borderwidth .code:n = {\def\AMCncol@BorderWidth{#1}},
1535 borderwidth .initial:n = { 1mm },
1536 backgroundcol .code:n = {\def\AMCncol@Background{#1}},
1537 backgroundcol .initial:n = { white },
1538 digits .int_set:N = \amc_num_ndigits_int,
1539 digits .initial:n = { 3 },
1540 decimals .int_set:N = \amc_num_decd_int,
1541 decimals .initial:n = { 0 },
1542 exponent .int_set:N = \amc_num_expo_int,
1543 exponent .initial:n = { 0 },
1544 base .int_set:N = \amc_num_base_int,
1545 base .initial:n = { 10 },
1546 sign .bool_set:N = \amc_num_sign_bool,
1547 sign .initial:n = { true },
1548 sign .default:n = { true },
1549 exposign .bool_set:N = \amc_num_exposign_bool,
1550 exposign .initial:n = { true },
1551 exposign .default:n = { true },
1552 strict .bool_set:N = \amc_num_strict_bool,
1553 strict .initial:n = { false },
1554 strict .default:n = { true },
1555 scoring .bool_set:N = \amc_num_scoring_bool,
1556 scoring .initial:n = { true },
1557 scoring .default:n = { true },
1558 ignoreblank .bool_set:N = \amc_num_ignoreblank_bool,
1559 ignoreblank .initial:n = { false },
1560 ignoreblank .default:n = { true },
1561 vertical .bool_set:N = \amc_num_vertical_bool,
1562 vertical .initial:n = { false },
1563 vertical .default:n = { true },
1564 expovertical .bool_set:N = \amc_num_expovertical_bool,
1565 expovertical .initial:n = { false },
1566 expovertical .default:n = { true },
1567 reverse .bool_set:N = \amc_num_reverse_bool,
1568 reverse .initial:n = { false },
1569 reverse .default:n = { true },
1570 vhead .bool_set:N = \amc_num_vhead_bool,
1571 vhead .initial:n = { false },
1572 vhead .default:n = { true },
1573 nozero .bool_set:N = \amc_num_nozero_bool,
1574 nozero .initial:n = { false },
1575 nozero .default:n = { true },
1576 significant .bool_set:N = \amc_num_significant_bool,
1577 significant .initial:n = { false },
1578 significant .default:n = { true },
1579 scoreexact .code:n = {\def\AMC@numeric@scoreexact{#1}},
1580 scoreexact .initial:n = { 2 },
1581 scoreapprox .code:n = {\def\AMC@numeric@scoreapprox{#1}},

```

```

1582 scoreapprox .initial:n = { 1 },
1583 scorewrong .code:n = {\def\AMC@numeric@scorewrong{#1}},
1584 scorewrong .initial:n = { 0 },
1585 exact .int_set:N = \amc_num_exact_int,
1586 exact .initial:n = { 0 },
1587 approx .int_set:N = \amc_num_approx_int,
1588 approx .initial:n = { 0 },
1589 keepas .code:n = {\def\AMC@numeric@keepas{#1}},
1590 keepas .initial:n = {},
1591 alsocorrect .code:n = {\def\AMC@numeric@alsocorrect{#1}},
1592 alsocorrect .initial:n = {}
1593 }
1594
1595 \cs_new:Npn \amc_num_setopts #1 {
1596   \keys_set:nn { amcnumeric } { #1 }
1597 }
1598
1599 \cs_new_nopar:Nn \amc_num_check_score_opts: {
1600   \bool_if:NTF \amc_num_ignoreblank_bool {
1601     \int_compare:nNnTF \amc_num_base_int = { 10 } { } {
1602       \message{^^J!~Error:~ignoreblank~can~only~be~used~with~number~base~10^^J}
1603     }
1604   } {}
1605 }
1606
1607 \cs_new_eq:NN \AMCnumericOpts \amc_num_setopts
1608

```

The command `\amc_num_char:nn{<i>}<j>}` draw a box with content *<inside>* (only if needed), where *<answer>* is `\AMC@checkedbox` if the corresponding choice is correct and empty if not.

```

1609 \cs_new:Npn \amc_num_char:nn #1 #2 {
1610   \global\advance\AMCrep@count\one\relax
1611   \AMCmessage{REP= \the\AMCrep@count :
1612     \ifx#2\AMC@checkedbox B\else M\fi }
1613   \ifAMC@correc
1614     \protect\AMC@keyBox@{#1}{#2}{1}{case : \AMCid@name :
1615       \the\AMCid@quest , \the\AMCrep@count}
1616   \else
1617     \protect\AMC@keyBox@{#1}{#2}{1}{case : \AMCid@name :
1618       \the\AMCid@quest , \the\AMCrep@count}
1619   \fi
1620 }

```

The command `\amc_num_digit_box:nn{<i>}<j>}` draws a box for current digit value *<i>*, where *<j>* is the correct value for the current digit. If *<i>* is greater than 9, it is converted to a character from the English alphabet (A for 10, B for 11...)

```

1621 \int_new:N \amc_num_digit_value_int
1622 \tl_new:N \amc_num_digit_value_tl
1623 \cs_new:Npn \amc_num_digit_box:nn #1 #2 {
1624   \int_set:Nn \amc_num_digit_value_int { #1 }
1625   \tl_set:Nn \amc_num_digit_value_tl {
1626     \int_compare:nNnTF { \amc_num_digit_value_int } < { 10 }
1627     { \int_to_arabic:n { \amc_num_digit_value_int } }

```

```

1628 { \int_to_Alph:n { \amc_num_digit_value_int - 9 } }
1629 }
1630 \int_compare:nNnTF { #1 } = { #2 } {
1631   \amc_num_char:nn{ \tl_use:N \amc_num_digit_value_tl }
1632     {\AMC@checkbox}
1633 } {
1634   \amc_num_char:nn{ \tl_use:N \amc_num_digit_value_tl }
1635     {}
1636 }
1637 }

```

The command `\amc_num_sign_boxes:Nn{<sign>}{<prefix>}` draws two boxes for the students to code the sign (with a right value given by the boolean `<negative>`).

```

1638 \cs_new:Npn \amc_num_sign_boxes:Nn #1 #2 {
1639   \int_case:nn { #1 } {
1640     { -1 } {
1641       \hbox{\amc_num_char:nn{ $+ $ }{}}
1642       \vspace{\AMCnumeric@Vspace}
1643       \AMCmessage{B=set. sign #2 =1}
1644       \hbox{\amc_num_char:nn{ $- $ }{\AMC@checkbox}}
1645       \AMCmessage{B=set. sign #2 =-1}
1646     }
1647     { 1 } {
1648       \hbox{\amc_num_char:nn{ $+ $ }{\AMC@checkbox}}
1649       \vspace{\AMCnumeric@Vspace}
1650       \AMCmessage{B=set. sign #2 =1}
1651       \hbox{\amc_num_char:nn{ $- $ }{}}
1652       \AMCmessage{B=set. sign #2 =-1}
1653     }
1654     { 0 } {
1655       \hbox{\amc_num_char:nn{ $+ $ }{}}
1656       \vspace{\AMCnumeric@Vspace}
1657       \AMCmessage{B=set. sign #2 =1}
1658       \hbox{\amc_num_char:nn{ $- $ }{}}
1659       \AMCmessage{B=set. sign #2 =-1}
1660     }
1661   }
1662 }

```

The command `\amc_num_digit_boxes_h:nnn{<varname>}{<correct>}{<maxdigit>}` draws a series of boxes for all possible values of a digit (from 0 to `<maxdigit>`), where the correct value is `<correct>`, transmitting scoring data to AMC so that the variable `<varname>` will be set to the value chosen by the student.

```

1663 \cs_new:Npn \amc_num_digit_boxes_h:nnn #1 #2 #3 {
1664   \int_step_inline:nnnn
1665   { \bool_if:NTF \amc_num_nozero_bool { 1 } { 0 } }
1666   { 1 } { #3 - 1 } {
1667     \amc_num_digit_box:nn { ##1 }{ #2 }
1668     \AMCmessage{B= set. #1 = ##1}
1669     \hspace{\AMCnumeric@Hspace}
1670   }
1671   \hspace{-\AMCnumeric@Hspace}
1672 }
1673

```

```

1674 \cs_new:Npn \amc_num_digit_boxes_v:nnn #1 #2 #3 {
1675   \int_step_inline:nnnn
1676   { \bool_if:NTF \amc_num_nozero_bool { 1 } { 0 } }
1677   { 1 } { #3 - 1 } {
1678     \vbox{\hbox{
1679       \amc_num_digit_box:nn { ##1 }{ #2 }
1680     }}
1681     \AMCmessage{B= set. #1 = ##1}
1682     \int_compare:nNnTF { ##1 } < { #3 - 1 } {
1683       \vspace{\AMCnumeric@Vspace}
1684     } {}
1685   }
1686 }
1687
1688 \int_new:N \amc_num_first_digit_int
1689 \cs_new:Npn \amc_num_digit_boxes_vr:nnn #1 #2 #3 {
1690   \int_set:Nn \amc_num_first_digit_int
1691   { \bool_if:NTF \amc_num_nozero_bool { 1 } { 0 } }
1692   \int_step_inline:nnnn { #3 - 1 } { -1 }
1693   \amc_num_first_digit_int {
1694     \vbox{\hbox{
1695       \amc_num_digit_box:nn { ##1 }{ #2 }
1696     }}
1697     \AMCmessage{B= set. #1 = ##1}
1698     \int_compare:nNnTF { ##1 } > \amc_num_first_digit_int {
1699       \vspace{\AMCnumeric@Vspace}
1700     } {}
1701   }
1702 }

```

The command `\amc_num_integer_boxes_v:Nnn{<correct digits>}{<prefix>}{<decimals>}` draws boxes for integer entry, without the sign.

```

1703 \cs_new:Npn \amc_num_integer_boxes_v:Nnn #1 #2 #3 {
  begin a loop over all digits,
1704   \int_set_eq:NN \amc_num_digit_int { \clist_count:N #1 }
1705   \clist_map_inline:Nn #1 {
    place the decimal point if necessary,
1706     \int_compare:nNnTF \amc_num_digit_int = { #3 } {
1707       \hbox{ \AMCdecimalPoint } \hspace{\AMCnumeric@Hspace}
1708     } { }
    draw the box for this digit,
1709     \hbox{\vbox{
1710       \bool_if:NTF \amc_num_vhead_bool {
1711         \vbox{\hbox{\AMCncontextVHead{ \int_eval:n
1712           { \amc_num_digit_int - 1 } }}}
1713         \vspace{\AMCnumeric@Vspace}
1714       } { }
1715       \bool_if:NTF \amc_num_reverse_bool {
1716         \amc_num_digit_boxes_vr:nnn { #2
1717           \int_to_Alph:n \amc_num_digit_int }
1718         { ##1 } { \amc_num_base_int }
1719       } {

```

```

1720      \amc_num_digit_boxes_v:nnn { #2
1721        \int_to_Alph:n \amc_num_digit_int }
1722      { ##1 } { \amc_num_base_int }
1723    }
1724  }}

```

and end the loop over digits, adding space if this is not the last one.

```

1725    \int_compare:nNnTF \amc_num_digit_int > 1 {
1726      \hspace{\AMCnumeric@Hspace}
1727    } { }
1728    \int_decr:N \amc_num_digit_int
1729  }
1730 }
1731

```

The command `\amc_num_integer_boxes_h:Nnn{<correct digits>}{<prefix>}{<decimals>}` does the same, in horizontal mode.

```

1732
1733 \cs_new:Npn \amc_num_integer_boxes_h:Nnn #1 #2 #3 {
1734   \vbox{
1735     \int_set_eq:NN \amc_num_digit_int { \clist_count:N #1 }
1736     \clist_map_inline:Nn #1 {
1737       \int_compare:nNnTF
1738         \amc_num_digit_int = { #3 } {
1739         \hbox{ \AMCdecimalPoint }
1740       } { }
1741     }
1742     \amc_num_digit_boxes_h:nnn { #2
1743       \int_to_Alph:n \amc_num_digit_int }
1744     { ##1 } \amc_num_base_int
1745   }
1746   \int_compare:nNnTF \amc_num_digit_int > 1 {
1747     \vspace{\AMCnumeric@Vspace}
1748   } { }
1749   \int_decr:N \amc_num_digit_int
1750 }
1751 }
1752

```

Finally, `\amc_num_integer_boxes:NnnNN{<correct digits>}{<prefix>}{<decimals>}{<sign bool>}{<sign>}` draws boxes for integer entry, including the sign if `<sign bool>` is true. When using the `strict` option, check the `+` box for a null value.

```

1753
1754 \cs_new:Npn \amc_num_integer_boxes:NnnNN #1 #2 #3 #4 #5 {
1755   \hbox{
1756     \bool_if:NTF { #4 } {
1757       \vbox{
1758         \ifx\AMCnTextSign\@empty\@empty\else
1759         \hbox{\AMCnTextSign}\vspace{\AMCnumeric@Vspace}\fi
1760         \bool_if:NTF \amc_num_strict_bool {
1761           \int_compare:nNnTF { #5 } = 0 {
1762             \amc_num_sign_boxes:Nn { 1 } { #2 }
1763           }{
1764             \amc_num_sign_boxes:Nn { #5 } { #2 }

```

```

1765     }
1766   }{
1767     \amc_num_sign_boxes:Nn { #5 } { #2 }
1768   }
1769 }
1770 \hspace{.5em}
1771 \vrule
1772 \hspace{.5em}
1773 } { }
1774 \hbox{
1775   \bool_if:NTF \amc_num_vertical_bool
1776   \amc_num_integer_boxes_v:Nnn \amc_num_integer_boxes_h:Nnn
1777   #1 { #2 } { #3 }
1778 }
1779 }
1780 }
1781

```

The command `\amc_num_build_integer_scoring:Nnnnn{<tl var>}{<sign bool>}{<prefix>}{<n>}{<decimals>}` builds a scoring to compute an integer from a serie of  $\langle n \rangle$ -digits boxes (from which  $\langle decimals \rangle$  are for decimals), with name prefix  $\langle prefix \rangle$ , using a sign variable if  $\langle sign bool \rangle$  is true.

```

1782
1783 \cs_new:Npn \amc_num_build_integer_scoring:Nnnnn #1 #2 #3 #4 #5 {
1784   \amc_num_check_score_opts:
1785   \tl_clear:N #1
1786   \int_set_eq:NN \amc_num_digit_int { #4 }
1787   \int_while_do:nNnn \amc_num_digit_int > 0 {
1788     \bool_if:NTF \amc_num_strict_bool {
1789       \AMCmessage{B=requires. #3
1790         \int_to_Alph:n \amc_num_digit_int = 1}
1791     } {
1792       \AMCmessage{B=default. #3
1793         \int_to_Alph:n \amc_num_digit_int =
1794         \bool_if:NTF \amc_num_ignoreblank_bool { } { 0 }
1795       }
1796     }
1797     \int_compare:nNnTF \amc_num_digit_int = #4 { } {
1798       \bool_if:NTF \amc_num_ignoreblank_bool {
1799         \tl_put_right:Nx #1 { ~.~ }
1800         \int_compare:nNnTF \amc_num_digit_int = #5 {
1801           \tl_put_right:Nx #1 { "." ~.~ }
1802         } { }
1803       } {
1804         \tl_put_left:Nn #1 { ( }
1805         \tl_put_right:Nx #1 { ) *
1806         \int_use:N \amc_num_base_int + }
1807       }
1808     }
1809     \tl_put_right:Nx #1
1810     { #3 \int_to_Alph:n \amc_num_digit_int }
1811     \int_decr:N \amc_num_digit_int
1812   }
1813   \bool_if:NTF \amc_num_ignoreblank_bool {
1814     \tl_put_left:Nn #1 { ( 0 + ( }

```

```

1815 \tl_put_right:Nn #1 { ) ) }
1816 \int_compare:nNnTF \amc_num_decd_int > 0 {
1817 \tl_put_right:Nx #1 { * ( 10 ** \int_eval:n { #5 } ) }
1818 } { }
1819 } {
1820 \tl_put_left:Nn #1 { ( }
1821 \tl_put_right:Nn #1 { ) }
1822 }
1823 \bool_if:NTF { #2 } {
1824 \bool_if:NTF \amc_num_strict_bool {
1825 \AMCmessage{B=requires. sign #3 =1}
1826 } {
1827 \AMCmessage{B=default. sign #3 =1}
1828 }
1829 \tl_put_right:Nx #1 { * ( sign #3 ) }
1830 } { }
1831 }
1832

```

Then the command `\AMCnumericShow{<x>}{<options>}` itself:

```

1833
1834 \fp_new:N \amc_num_result_fp
1835 \fp_new:N \amc_num_correct_fp
1836 \clist_new:N \amc_num_digits_clist
1837 \clist_new:N \amc_num_expo_digits_clist
1838 \int_new:N \amc_num_digit_int
1839 \int_new:N \amc_num_sign_int
1840 \int_new:N \amc_num_expo_sign_int
1841 \tl_new:N \amc_num_compute_tl
1842 \tl_new:N \amc_num_expo_tl
1843 \int_new:N \amc_num_correct_expo_int
1844
1845 \cs_new:Npn \amc_numeric_show:nn #1 #2 {

```

We have to tell AMC that the scoring we will give concerns this question:

```

1846 \ifAMC@ensemble\ifAMCformulaire@dedans
1847 \AMCmessage{Q=\the\AMCid@quest}
1848 \fi\fi

```

Then we parse the options from `<opts>`:

```

1849 { \keys_set:nn { amcnumeric } { #2 }
1850 \bool_if:nTF { \bool_if_p:N\amc_num_significant_bool
1851 && \int_compare_p:n { \amc_num_base_int != 10 } } {
1852 \message{^^J!~AMCnumeric~Error:~significant=true~can't~be~used~with~base!=10.^^J}
1853 } { }
1854 \bool_if:nTF { \int_compare_p:n { \amc_num_expo_int != 0 }
1855 && \int_compare_p:n { \amc_num_base_int != 10 } } {
1856 \message{^^J!~AMCnumeric~Error:~scientific~notation~can't~be~used~with~base!=10.^^J}
1857 } { }

```

Convert the floating point correct value to integer, taking into account the parameters significant, exponent and decimals:

```

1858 \ifx\@empty#1\@empty
1859 \fp_set:Nn \amc_num_correct_fp { 0 }
1860 \fp_set:Nn \amc_num_mantissa_fp { 0 }

```

```

1861     \int_set:Nn \amc_num_correct_expo_int { 0 }
1862   \else
1863     \bool_if:NTF \amc_num_significant_bool {
1864       \amc_fp_n_significant_digits:Nnn \amc_num_correct_fp { #1 } \amc_num_ndigits_int
1865     } {
1866       \int_compare:nNnTF \amc_num_expo_int > 0 {
1867         \amc_fp_decompose:Nnn \amc_num_mantissa_fp \amc_num_correct_expo_int { #1 }
1868         \int_compare:nNnTF { \amc_num_ndigits_int - \amc_num_decd_int } > 1 {
1869           \fp_set:Nn \amc_num_mantissa_fp {
1870             \amc_num_mantissa_fp * 10^( \amc_num_ndigits_int - \amc_num_decd_int - 1 )
1871           }
1872           \int_set:Nn \amc_num_correct_expo_int {
1873             \amc_num_correct_expo_int - ( \amc_num_ndigits_int - \amc_num_decd_int - 1 )
1874           }
1875         } {}
1876         \amc_fp_n_digits:Nnn \amc_num_correct_fp \amc_num_mantissa_fp \amc_num_decd_int
1877       } {
1878         \amc_fp_n_digits:Nnn \amc_num_correct_fp { #1 } \amc_num_decd_int
1879       }
1880     }
1881   \fi

```

Now extracts the required digits:

```

1882   \ifx\@empty#1\@empty
1883     \amc_invalid_digits:Nn \amc_num_digits_clist \amc_num_ndigits_int
1884     \amc_invalid_digits:Nn \amc_num_expo_digits_clist \amc_num_expo_int
1885     \int_set:Nn \amc_num_sign_int { 0 }
1886     \int_set:Nn \amc_num_expo_sign_int { 0 }
1887   \else
1888     \amc_fp_to_digits:Nnnn \amc_num_digits_clist \amc_num_correct_fp
1889     \amc_num_ndigits_int \amc_num_base_int
1890     \amc_get_fp_sign:Nn \amc_num_sign_int \amc_num_correct_fp
1891     \int_compare:nNnTF \amc_num_expo_int > 0 {
1892       \amc_fp_to_digits:Nnnn \amc_num_expo_digits_clist \amc_num_correct_expo_int
1893       \amc_num_expo_int \amc_num_base_int
1894       \amc_get_int_sign:Nn \amc_num_expo_sign_int \amc_num_correct_expo_int
1895     } {}
1896   \fi

```

The question scoring is given to AMC (if requested by the `scoring=true` option). Note that the variable `intV` refers to the correct value, and `intX` to the value entered by the student.

```

1897   \fp_set:Nn \amc_num_result_fp { #1 }
1898   \AMCmessage{B=numval=\fp_to_scientific:N \amc_num_result_fp ,
1899     numex=\int_use:N \amc_num_exact_int,
1900     numapp=\int_use:N \amc_num_approx_int,
1901     numsex=\AMC@numeric@scoreexact,
1902     numsapp=\AMC@numeric@scoreapprox
1903   }
1904   \bool_if:NTF \amc_num_scoring_bool {
1905     \AMCmessage{B=haut=mz=d=undef,p=undef,
1906       formula=(Vdifference <= \int_use:N \amc_num_exact_int ?
1907         \AMC@numeric@scoreexact :
1908         \int_compare:nNnTF \amc_num_approx_int = 0 {
1909           \AMC@numeric@scorewrong

```



```

1910     } {
1911         (Vdifference <= \int_use:N\amc_num_approx_int ?
1912         \AMC@numeric@scoreapprox : \AMC@numeric@scorewrong)
1913     }
1914 }}
1915 } {}
1916 \amc_num_build_integer_scoring:Nnnnn
1917   \amc_num_compute_tl \amc_num_sign_bool { digit } \amc_num_ndigits_int
1918   \amc_num_decd_int
1919   \int_compare:nNnTF \amc_num_expo_int > 0 {
1920       \amc_num_build_integer_scoring:Nnnnn
1921       \amc_num_expo_tl \amc_num_exposign_bool { expo } \amc_num_expo_int { 0 }
1922       \AMCmessage{B= set. intE = \amc_num_expo_tl}
1923   } {}
1924   \AMCmessage{B= set.intV = \fp_to_int:N\amc_num_correct_fp ,
1925       set.intXX = \amc_num_compute_tl }
1926   \int_compare:nNnTF \amc_num_expo_int > 0 {
1927       \AMCmessage{B= set.intX = intXX * \int_use:N\amc_num_base_int ** (intE - (\int_use:N\amc_num_correct
1928   ){
1929       \AMCmessage{B= set.intX = intXX}
1930   }
1931   \int_compare:nNnTF \amc_num_expo_int > 0 {
1932       \AMCmessage{B= set.valueX = intXX * \int_use:N\amc_num_base_int ** (intE - \int_use:N\amc_num_decd_int)}
1933   ){
1934       \AMCmessage{B= set.valueX = intXX * \int_use:N\amc_num_base_int ** (- \int_use:N\amc_num_decd_int)}
1935   }
1936   \ifx\@empty\AMC@numeric@keepas\@empty\else
1937       \AMCmessage{B= setglobal.\AMC@numeric@keepas = valueX}
1938   \fi
1939   \ifx\@empty#1\@empty
1940       \bool_if:NTF \amc_num_significant_bool {
1941           \AMCmessage{B=set.Vdifference=0}
1942       ){
1943           \ifx\@empty\AMC@numeric@alsocorrect\@empty
1944               \AMCmessage{B=set.Vdifference=0}
1945           \else
1946               \AMCmessage{B="set.Vdifference =
1947                   amcvdifference( \AMC@numeric@alsocorrect, valueX, \int_use:N\amc_num_decd_int, \int_use:N\amc_num
1948                   "}
1949           \fi
1950       }
1951   \else
1952       \bool_if:NTF \amc_num_significant_bool {
1953           \AMCmessage{B=set.Vdifference="min( abs((intV)-(intX)) ,
1954               abs(\int_use:N\amc_num_base_int * (intV) - (intX)) ,
1955               abs((intV) - \int_use:N\amc_num_base_int * (intX)) )"}
1956       } {
1957           \ifx\@empty\AMC@numeric@alsocorrect\@empty
1958               \AMCmessage{B=set.Vdifference=abs((intV)-(intX))}
1959           \else
1960               \AMCmessage{B="set.Vdifference =
1961                   min( amcvdifference( \AMC@numeric@alsocorrect, valueX, \int_use:N\amc_num_decd_int, \int_use:N\amc
1962                   abs((intV)-(intX)) )"}

```

```

1963   \fi
1964 }
1965 \fi

Begin now with the frame around all the boxes:

1966 \ifAMC@extractOnly\else
1967 \vspace{1.5ex}\par{
1968   \fboxrule=\AMCncol@BorderWidth
1969   \fcolorbox{\AMCncol@Border}{\AMCncol@Background}{
1970     \bool_if:NTF \amc_num_expovetical_bool {
1971       \hbox{\vbox{
1972         \vbox{\amc_num_integer_boxes:NnnNN
1973           \amc_num_digits_clist { digit } \amc_num_decd_int \amc_num_sign_bool
1974           \amc_num_sign_int}
1975         \int_compare:nNnTF \amc_num_expo_int > 0 {
1976           \vspace{\AMCnumeric@Vspace}
1977           \vbox{\hbox{\AMCexponent}}
1978           \vspace{\AMCnumeric@Vspace}
1979           \vbox{\amc_num_integer_boxes:NnnNN
1980             \amc_num_expo_digits_clist { expo } { 0 } \amc_num_exposign_bool
1981             \amc_num_expo_sign_int}
1982           } {}
1983         }}
1984       } {
1985         \amc_num_integer_boxes:NnnNN
1986         \amc_num_digits_clist { digit } \amc_num_decd_int \amc_num_sign_bool
1987         \amc_num_sign_int
1988         \int_compare:nNnTF \amc_num_expo_int > 0 {
1989           \hspace{\AMCnumeric@Hspace}\AMCexponent\hspace{\AMCnumeric@Hspace}
1990           \amc_num_integer_boxes:NnnNN
1991           \amc_num_expo_digits_clist { expo } { 0 } \amc_num_exposign_bool
1992           \amc_num_expo_sign_int
1993         } {}
1994       }
1995     }
1996   }
1997 \fi

```

And tell AMC that we finished with this question:

```

1998 \ifAMC@ensemble\else\vspace{1.5ex}\par\fi
1999 \ifAMC@ensemble\ifAMCformulaire@dedans
2000   \AMCmessage{FQ}
2001 \fi\fi
2002 }
2003 }
2004
2005 \cs_new_eq:NN \AMCnumericShow \amc_numeric_show:nn
2006

```

`\AMCnumericHide` is called when the boxes are not to be drawn (in the question sheets for separate answer sheet layout), and `\AMCnumericChoices{⟨value⟩}{⟨options⟩}` is the function to be used in the LaTeX source code of the exam.

```

2007 \cs_new:Npn \amc_numeric_hide:nn #1 #2 {
2008   \keys_set:nn { amcnumeric } { #2 }

```

```

2009 \AMCnTextGoto
2010 \ifAMC@qbloc\else\vspace{1.5ex}\par\fi
2011 }
2012
2013 \cs_new_eq:NN \AMCnumericHide \amc_numeric_hide:nn
2014
2015 \ExplSyntaxOff
2016 \def\AMCnumericChoicesPlain{%
2017   \AMC@if@separate@question{\AMC@mem@category{numeric}}%
2018   \AMCformatChoices{\AMCnumericShow}{\AMCnumericHide}%
2019 }

```

The  $\langle value \rangle$  argument is often given as a macro, that is to be expanded before calling `\AMCnumericChoicesPlain`, so that its value will be the same in the separate answer sheet...

```

2020 \ExplSyntaxOn
2021
2022 \cs_new:Npn \amc_numeric_choices:nn #1#2 {
2023   \AMCnumericChoicesPlain{#1}{#2}
2024 }
2025 \cs_generate_variant:Nn \amc_numeric_choices:nn { xn }
2026 \cs_new_eq:NN \AMCnumericChoices \amc_numeric_choices:xn
2027
2028 \ExplSyntaxOff

```

### 4.13.3 Intervals

`\AMCIntervals` The command `\AMCIntervals{\langle x \rangle}{\langle x0 \rangle}{\langle x1 \rangle}{\langle \delta \rangle}` can be used to present answers as intervals  $[x_i, x_i + \delta[$  covering  $[\langle x0 \rangle, \langle x1 \rangle]$ , such that the only interval containing  $\langle x \rangle$  is declared as `\correctchoice`, and the other as `\wrongchoice`.

For this command to work, one has to load package `fp`.

As an example,

```

\begin{question}{quarter}
  In which interval falls  $1/4$ ?
  \begin{multicols}{5}
    \begin{choices}[o]
      \AMCIntervals{0.25}{0}{1}{0.1}
    \end{choices}
  \end{multicols}
\end{question}

```

produces (in correction mode):

**Question 12** In which interval falls  $1/4$ ?

<input type="checkbox"/> $[0, 0.1[$	<input checked="" type="checkbox"/> $[0.2, 0.3[$	<input type="checkbox"/> $[0.4, 0.5[$	<input type="checkbox"/> $[0.6, 0.7[$	<input type="checkbox"/> $[0.8, 0.9[$
<input type="checkbox"/> $[0.1, 0.2[$	<input type="checkbox"/> $[0.3, 0.4[$	<input type="checkbox"/> $[0.5, 0.6[$	<input type="checkbox"/> $[0.7, 0.8[$	<input type="checkbox"/> $[0.9, 1[$

Note that the interval formatting can be changed redefining the `\AMCintervalFormat` command, which is originally defined as

```

2029 \def\AMCIntervalFormat#1#2{[#1,\,#2]}

```

to follow local conventions (writting  $[a, b]$  instead of  $[a, b[$  is for example a common usage).

```

2030 \ExplSyntaxOn
2031
2032 \fp_new:N \amc_interv_a
2033 \fp_new:N \amc_interv_b
2034 \cs_new:Npn \amc_intervals:nnnn #1 #2 #3 #4 {
2035   \fp_set:Nn \amc_interv_a { #2 }
2036   \fp_do_while:nn { \amc_interv_a < #3 } {
2037     \fp_set:Nn \amc_interv_b { \amc_interv_a + #4 }
2038     \fp_compare:nTF { \amc_interv_a <= #1 < \amc_interv_b }
2039       \correctchoice \wrongchoice
2040       {\AMCIntervalFormat{\fp_use:N \amc_interv_a}{\fp_use:N \amc_interv_b}}
2041     \fp_set:Nn \amc_interv_a \amc_interv_b
2042   }
2043 }
2044 \cs_new_eq:NN \AMCIntervals \amc_intervals:nnnn
2045
2046 \ExplSyntaxOff

```

## 4.14 Open questions

`\AMCOpen` The command `\AMCOpen{<options>}{<choices>}` can be used as a replacement for the `choices` environment when asking the student to write some answer by hand. The teacher will correct and mark this answer either on the paper before scanning, or with manual data capture, thanks to the scoring boxes.

As an example,

```

\begin{question}{Linux}
  What is the first name of the person who started working on the Linux kernel?
  \AMCOpen{}{\wrongchoice[w]{w}\scoring{0}\correctchoice[c]{c}\scoring{2}}
\end{question}

```

shows:

<b>Question 13</b> What is the first name of the person who started working on the Linux kernel? <div style="text-align: right; margin-top: -10px;"> <input type="checkbox"/> w   <input type="checkbox"/> c </div> <div style="border: 1px solid black; height: 40px; margin-top: 10px; position: relative;"> <div style="position: absolute; bottom: 5px; left: 5px;"> ..... </div> </div>
--

The teacher will have to tick the ‘w’ box for wrong answers, and the ‘c’ box for correct answers.

Begin with the options definitions:

```

2047 \def\AMCotextGoto{}
2048 \def\AMCotextReserved{}
2049 \def\AMCocol@Background{lightgray}
2050 \def\AMCocol@BoxFrameRule{white}
2051 \def\AMCocol@FrameRule{black}
2052 \def\AMCocol@Foreground{}
2053 \def\AMCopen@answer{}
2054 \def\AMCopen@question{}

```

```

2055 \def\AMCopen@lineuptext{}
2056 \define@key{AMCOpen}{backgroundcol}{\def\AMCocol@Background{#1}}
2057 \define@key{AMCOpen}{foregroundcol}{\def\AMCocol@Foreground{#1}}
2058 \define@key{AMCOpen}{Treserved}{\def\AMCotextReserved{#1}}
2059 \define@key{AMCOpen}{question}{\AMCid@name}{\def\AMCopen@question{#1}}
2060 \define@key{AMCOpen}{answer}{\def\AMCopen@answer{#1}}
2061 \define@key{AMCOpen}{contentcommand}[AMCopen@lines]{\def\AMCopen@contentcommand{#1}}
2062 \newdimen\AMCopen@Hspace\AMCopen@Hspace=.5em
2063 \define@key{AMCOpen}{hspace}{\AMCopen@Hspace=#1}
2064 \def\AMCopen@Width{.95\linewidth}
2065 \define@key{AMCOpen}{width}{\def\AMCopen@Width{#1}}
2066 \newdimen\AMCopen@LineHeight\AMCopen@LineHeight=1cm
2067 \define@key{AMCOpen}{lineheight}{\AMCopen@LineHeight=#1}
2068 \newcount\AMCopen@Lines\AMCopen@Lines=1
2069 \define@key{AMCOpen}{lines}{\AMCopen@Lines=#1}
2070 \newdimen\AMCopen@boxmargin\AMCopen@boxmargin=3pt
2071 \define@key{AMCOpen}{boxmargin}{\AMCopen@boxmargin=#1}
2072 \newdimen\AMCopen@boxframerule\AMCopen@boxframerule=1pt
2073 \define@key{AMCOpen}{boxframerule}{\AMCopen@boxframerule=#1}
2074 \define@key{AMCOpen}{boxframerulecol}{\def\AMCocol@BoxFrameRule{#1}}
2075 \define@key{AMCOpen}{framerulecol}{\def\AMCocol@FrameRule{#1}}
2076 \newdimen\AMCopen@framerule\AMCopen@framerule=1pt
2077 \define@key{AMCOpen}{framerule}{\AMCopen@framerule=#1}
2078 \define@key{AMCOpen}{lineuptext}{\def\AMCopen@lineuptext{#1}}
2079 \define@boolkey{AMCOpen}{dots}[true]{}
2080 \define@boolkey{AMCOpen}{scan}[true]{}
2081 \define@boolkey{AMCOpen}{retick}[true]{}
2082 \define@boolkey{AMCOpen}{annotate}[false]{}
2083 \define@boolkey{AMCOpen}{lineup}[false]{}
2084 \setkeys{AMCOpen}{dots,scan,retick,annotate,lineup,contentcommand}
2085 \newcommand\AMCopenOpts[1]{\setkeys{AMCOpen}{#1}}

```

The command `\AMCOpen` is similar to `\AMCnumericChoices`, calling either `\AMCopenShow` or `\AMCopenHide`.

```

2086 \newcommand\AMCopen@lines{%
2087   \begin{minipage}{\AMCopen@Width}%
2088     \loop\vspace{\AMCopen@LineHeight}
2089     \hspace*{.5em}\ifAMC@correc\smash{\AMCopen@answer}\def\AMCopen@answer{}\fi%
2090     \ifKV@AMCopen@dots%
2091       \dotfill\hspace*{.5em}
2092     \fi
2093     \ifnum\AMCopen@Lines>\@ne\par\advance\AMCopen@Lines\m@ne\repeat%
2094   \end{minipage}
2095 }
2096 \newcommand\AMCopenShow[2]{
2097   \ifAMC@ensemble\ifAMCformulaire@dedans%
2098     \AMCmessage{Q=\the\AMCid@quest}%
2099   \fi\fi%
2100   {\setkeys{AMCOpen}{#1}}%
2101   \ifKV@AMCopen@lineup%
2102     \ifAMC@ensemble\else%
2103       \ifx\@empty\AMCopen@lineuptext\@empty\fi%
2104     \fi%
2105     \ifAMC@correc\smash{\AMCopen@answer}\fi%

```

```

2106 \ifx\@empty\AMCopen@lineuptext\@empty%
2107 \dotfill%
2108 \else%
2109 \AMCopen@lineuptext\hfill%
2110 \fi%
2111 \else%
2112 \hspace*{.5em}\linebreak[1]\hspace*{\fill}%
2113 \fi%
2114 {\AMCnoCompleteMulti%
2115 \def\AMCbeginAnswer{}\def\AMCendAnswer{}%
2116 \def\AMCanswer##1##2{\ifAMC@ensemble ##1\else%
2117 \ifAMC@inside@box ##1\else{\AMCboxOutsideLetter{##1}{##2}}\fi\fi%
2118 \hspace{\AMCopen@Hspace}}%
2119 \fboxsep=\AMCopen@boxmargin%
2120 \fboxrule=\AMCopen@boxframerule%
2121 \fcolorbox{\AMCocol@BoxFrameRule}{\AMCocol@Background}{%
2122 \ifAMC@ensemble\AMCopen@question%
2123 \ifx\@empty\AMCopen@question\@empty\else\hspace{\AMCopen@Hspace}\fi%
2124 \fi%
2125 \begin{choicescustom}[o]%
2126 \ifx\AMCocol@Foreground\@empty\@empty\else%
2127 \def\AMC@boxcolor{\AMCocol@Foreground}%
2128 \fi%
2129 #2%
2130 \ifKV@AMCOpen@scan\else\AMCdontScan\fi%
2131 \ifKV@AMCOpen@retick\AMCreTick\fi%
2132 \ifKV@AMCOpen@annotate\else\AMCdontAnnotate\fi%
2133 \end{choicescustom}%
2134 \ifx\@empty\AMCotextReserved\@empty%
2135 \hspace{-\AMCopen@Hspace}%
2136 \else%
2137 \ifx\AMCocol@Foreground\@empty\@empty%
2138 \AMCotextReserved%
2139 \else%
2140 \textcolor{\AMCocol@Foreground}{\AMCotextReserved}%
2141 \fi%
2142 \fi%
2143 }%
2144 \ifKV@AMCOpen@lineup\else%
2145 \par\nobreak\noindent%
2146 \hspace*{\fill}%
2147 \fboxrule=\AMCopen@framerule%
2148 \fcolorbox{\AMCocol@FrameRule}{white}{%
2149 \cname\AMCopen@contentcommand\endcname
2150 }%
2151 \vspace{\AMCpostOquest}\par%
2152 \fi%
2153 }%
2154 \ifAMC@ensemble\ifAMCformulaire@dedans%
2155 \AMCmessage{FQ}%
2156 \fi\fi%
2157 }
2158 \newcommand\AMCopenHide[2]{%

```

```

2159 \AMCotextGoto%
2160 \ifAMC@qbloc\else\vspace{1.5ex}\par\fi%
2161 }
2162 \def\AMCOpen{%
2163 \AMC@if@separate@question{\AMC@mem@category{open}}%
2164 \AMCformatChoices{\AMCopenShow}{\AMCopenHide}%
2165 }

```

#### 4.15 Boxes with letters only

\AMCBoxOnly Sometimes the letters printed in the boxes (or just after them) are enough to describe the answers. In such cases, printing the boxes both on the question and on the answer sheet is not necessary. The \AMCBoxOnly{<options>}{<choices>} can be used as a replacement for the choices environment:

```

\begin{question}{arm}
  Which letter shows the \textit{arm} on the diagram?
  \AMCBoxOnly{ordered=true}{\wrongchoice[A]{ }\correctchoice[B]{ }%
    \wrongchoice[C]{ }\wrongchoice[D]{ }}
\end{question}

2166 \def\AMCbotextGoto{}
2167 \def\AMCbo@help{}
2168 \define@key{AMCBoxOnly}{help}{\def\AMCbo@help{#1}}
2169 \define@boolkey{AMCBoxOnly}{ordered}[false]{ }
2170 \setkeys{AMCBoxOnly}{ordered}
2171 \newcommand\AMCboOpts[1]{\setkeys{AMCBoxOnly}{#1}}
2172 \newcommand\AMCboShow[2]{%
2173 \ifAMC@ensemble\ifAMCformulaire@dedans%
2174 \AMCmessage{Q=\the\AMCid@quest}%
2175 \fi\fi%
2176 {\setkeys{AMCBoxOnly}{#1}%
2177 \def\AMCbeginAnswer{}\def\AMCendAnswer{}%
2178 \def\AMCanswer##1##2{\hspace{\AMCformHSpace} \ifAMC@ensemble ##1\else%
2179 \ifAMC@inside@box ##1\else\AMCboxOutsideLetter{##1}{##2}}\fi\fi%
2180 }%
2181 \ifAMC@ensemble\AMCbo@help\fi%
2182 \ifKV@AMCBoxOnly@ordered%
2183 \begin{choicescustom}[o]%
2184 \else%
2185 \begin{choicescustom}%
2186 \fi%
2187 #2
2188 \end{choicescustom}%
2189 }%
2190 \ifAMC@ensemble\ifAMCformulaire@dedans%
2191 \AMCmessage{FQ}%
2192 \fi\fi%
2193 }
2194 \newcommand\AMCboHide[2]{
2195 \AMCbotextGoto%
2196 \ifAMC@qbloc\else\vspace{1.5ex}\par\fi%
2197 }

```

```

2198 \def\AMCBoxOnly{%
2199   \AMC@if@separate@question{\AMC@mem@category{box}}%
2200   \AMC@formatChoices{\AMCboShow}{\AMCboHide}%
2201 }

```

## 4.16 Page formatting

### 4.16.1 Watermark

`\AMCw@termark` These commands are used to print a grey “DRAFT” under each page, so as to prevent from  
`\AMCw@terprint` printing old versions of the subject.

```

2202 \DeclareFontShape{OT1}{cmr}{b}{n}{<35->cmr17}{-}
2203 \def\AMC@watertext{\AMC@loc@draft}
2204 \newcommand\AMCw@termark{%
2205   \setlength{\@tempdimb}{.5\paperwidth}%
2206   \setlength{\@tempdimc}{-.5\paperheight}%
2207   \put(\strip@pt\@tempdimb,\strip@pt\@tempdimc){%
2208     \makebox(0,0){\rotatebox{45}{\AMC@LR{%
2209       \textcolor[gray]{0.8}{
2210         \fontencoding{OT1}\fontfamily{cmr}
2211         \fontseries{b}\fontshape{n}
2212         \fontsize{90pt}{120pt}
2213         \selectfont
2214         \AMC@watertext}}}}}
2215 \newcommand\AMCw@terprint[1]{%
2216   \setbox\@tempboxa\vbox to \z@{%
2217     \vbox{%
2218       \hbox to \z@{%
2219         #1\hss}\vss}
2220   \dp\@tempboxa\z@
2221   \box\@tempboxa}

```

### 4.16.2 Signs for scan analysis

The following code sets up all the signs to be printed on the pages so as to be able to recognize the position of the boxes on the scans. Four circles ● are printed on the corners (see `\m@rqueCalage`), and binary boxes show the student sheet number (see `\AMCIDBoxesA`), the page (see `\AMCIDBoxesB`) and a checking number (see `\AMCIDBoxesC`).

`\AMC@intituleHead` is the title to be printed at the beginning (used for corrected sheet, and empty on subject). `\AMC@note` is printed at the bottom of each page. You can change its value using `\AMCsetFoot{<foot>}`.

```

2222 \def\AMCcircle#1#2{%
2223   {\setlength{\unitlength}{1mm}%
2224     \begin{picture}(\#1,\#1)(-\#2,-\#2)\thinspace\circle*{\#1}\end{picture}}}
2225 \def\m@rqueCalage{\AMCcircle{3.6}{1.8}}
2226 \def\m@rque#1{\AMC@tracebox{1}{\#1}{\m@rqueCalage}}
2227 \def\he@dtaille#1{% \par cancels the \leavevmode
2228   % introduced by https://github.com/pietvo/fancyhdr/commit/6b1ad10eeb5bc3d804f3cd2cf193e6440d0229e6
2229   \par\vbox to 1cm{\#1}}
2230 \def\he@dbas#1{\he@dtaille{\vspace*{\fill}\#1}}
2231 \def\he@dhaut#1{\he@dtaille{\#1\vspace*{\fill}}}
2232 \def\AMC@intituleHead{\AMC@loc@corrected}

```



```

2233 \def\AMC@note{}
2234 \def\AMCsetFoot#1{\def\AMC@note{#1}}
2235 \newcommand\AMCStudentNumber{\the\AMCid@etud}
2236 \newcommand\AMCIDBoxesA{\AMC@binaryCode{id=1,ndigits=\AMC@NCBetud}{\the\AMCid@etud}}
2237 \newcommand\AMCIDBoxesB{\AMC@binaryCode{id=2,ndigits=\AMC@NCBpage}{\thepage}}
2238 \newcommand\AMCIDBoxesC{\AMC@binaryCode{id=3,ndigits=\AMC@NCBcheck}{\the\AMCid@check}}
2239 \newcommand\AMCIDBoxesABC{%
2240   \hbox{\vbox{\noindent\AMCIDBoxesA\
2241     \noindent\AMCIDBoxesB\AMCIDBoxesC}}}%
2242 }
2243 \AtBeginPage{\ifAMC@pagelayout\global\advance\AMCid@check\m@ne%
2244   \ifnum\AMCid@check<1\global\AMCid@check=\AMCid@checkmax\fi%
2245   \AMC@pagepos%
2246   \ifAMC@watermark\ifAMC@correchad\else\AMCw@terprint{\AMCw@termark}%
2247   \fi\fi\fi}
2248 \fancypagestyle{AMCpageHeadOnly}{%
2249   \fancyhf{}\fancyhead[C]{\textsc{\AMC@intituleHead}}%
2250   \renewcommand{\headrulewidth}{0pt}%
2251   \renewcommand{\footrulewidth}{0pt}%
2252 }
2253 \fancypagestyle{AMCpageFull}{%
2254   \fancyhf{}%
2255   \fancyhead[L]{\AMC@LR{\he@dbas{\leavevmode\m@rque{positionHG}}}}%
2256   \fancyhead[R]{\AMC@LR{\he@dbas{\leavevmode\m@rque{positionHD}}}}%
2257   \fancyfoot[L]{\AMC@LR{\leavevmode\m@rque{positionBG}}}%
2258   \fancyfoot[R]{\AMC@LR{\leavevmode\m@rque{positionBD}}}%
2259   \fancyhead[C]{\AMC@LR{\he@dhaut{%
2260     \begin{minipage}[b]{\AMC@CBtaille}\AMCboxColor{black}%
2261       \ifAMCids@top\vbox to \AMCids@height{\texttt{+\the\AMCid@etud/\thepage/\the\AMCid@check+}}\fi%
2262       \AMCIDBoxesABC
2263     \end{minipage}%
2264     \ifAMCids@side\hbox to \AMCids@width{\hspace*{\fill}%
2265       \texttt{+\the\AMCid@etud/\thepage/\the\AMCid@check+}}\fi%
2266     }}}%
2267   \fancyhfoffset[EOLR]{5mm}%
2268   \fancyfoot[C]{\AMC@note}%
2269   \renewcommand{\headrulewidth}{0pt}%
2270   \renewcommand{\footrulewidth}{0pt}%
2271 }
2272 \newcommand\AMCsubjectPageTag{%
2273   \fbox{\texttt{\the\AMCid@etud:\thepage}}%
2274 }
2275 \fancypagestyle{AMCpageNoMarks}{%
2276   \fancyhf{}%
2277   \fancyhead[R]{\AMCsubjectPageTag}%
2278   \fancyfoot[C]{\AMC@note}%
2279   \renewcommand{\headrulewidth}{0pt}%
2280   \renewcommand{\footrulewidth}{0pt}%
2281 }
2282 \fancypagestyle{AMCpageEmpty}{%
2283   \fancyhf{}%
2284   \renewcommand{\headrulewidth}{0pt}%
2285   \renewcommand{\footrulewidth}{0pt}%

```

```

2286 }
2287 \AtBeginDocument{%
2288   \ifAMC@pagelayout%
2289     \ifAMC@correchead
2290       \pagestyle{AMCpageHeadOnly}
2291     \else
2292       \pagestyle{AMCpageFull}
2293     \fi
2294   \fi
2295 }

```

#### 4.17 Defining a single exam copy content

`\onecopy` The command `\onecopy[ $\langle n \rangle$ ]{ $\langle code \rangle$ }` generates  $\langle n \rangle$  copies of the subject that is described in  $\langle code \rangle$ . The L<sup>A</sup>T<sub>E</sub>X code  $\langle code \rangle$  that generates a single copy can be a little long, so that the environment `examcopy` is often preferred.

```

2296 \newcommand{\onecopy}[2]{%
2297   \ifx\AMCNombreCopies\undefined\AMCnum@copies=#1%
2298   \else\AMCnum@copies=\AMCNombreCopies\fi%
2299   \AMCmessage{TOTAL=\the\AMCnum@copies}%
2300   \AMCid@etud=\AMCid@etudstart%
2301   \ifnum\AMCid@etud=0\AMCid@etud=\AMC@premierecopie\fi%
2302   \AMCid@etudfin=\AMCnum@copies%
2303   \advance\AMCid@etudfin\AMCid@etud\relax%
2304   \ifAMC@correchead\AMCid@etudfin=\AMC@premierecopie\fi
2305   \ifAMC@pdfform\begin{Form}\fi%
2306   \loop{%
2307     \ifAMC@calibration\protected@write\AMC@XYFILE{}{%
2308       \string\rngstate{\the\AMCid@etud}\the\AMC@SR}%
2309     }\fi%
2310     \AMC@zoneformulairefalse\setcounter{page}{1}\setcounter{section}{0}%
2311     \ifAMC@ensemble\ifAMC@automarks\pagestyle{AMCpageNoMarks}\fi\fi%
2312     \AMCnumero{1}%
2313     \ifAMC@calibration\AMCmessage{ETU=\the\AMCid@etud}\fi%
2314     \AMC@multiclear%
2315     \global\AMC@keepmemoryfalse%
2316     #2%
2317     \ifAMC@keepmemory\else\AMC@mem@clear\fi%
2318     \clearpage}%
2319   \advance\AMCid@etud\@ne\ifnum\AMCid@etud<\AMCid@etudfin\repeat%
2320   \global\AMCid@etudstart=\AMCid@etud%
2321   \ifAMC@pdfform\end{Form}\fi%
2322   \AMC@multi@report%
2323 }

```

`\AMCaddpagesto` In some situations, one needs all question sheets to have the same number of pages. The command `\AMCaddpagesto{ $\langle n \rangle$ }` adds enough (white) pages to get at least  $\langle n \rangle$  pages in the current question sheet.

```

2324 \newcount\AMC@addpages
2325 \newcommand{\AMCaddpagesto}[1]{%
2326   \AMC@addpages=#1\advance\AMC@addpages\@ne%
2327   \clearpage%

```

```

2328 \@whilenum\thepage<\AMC@addpages\do{%
2329     \ifAMC@automarks\pagestyle{AMCpageEmpty}\fi%
2330     \hbox{}\clearpage%
2331 }%
2332 }

```

**\AMCcleardoublepage** If you want to print the subject all at one time in duplex mode, it is necessary to end each subject with an even number of pages. This can be achieved using **\AMCcleardoublepage** at the end of the copy definition. This command is also useful inserted before the separate answer sheet (if any).

```

2333 \def\AMCcleardoublepage{%
2334     \clearpage%
2335     \ifodd\thepage\else%
2336         \ifAMC@automarks\pagestyle{AMCpageEmpty}\fi%
2337         \hbox{}\clearpage%
2338     \fi%
2339 }

```

**\exemplairepair** To make some differences in the copies, checking if the student sheet number is odd, with **\exemplairepair** construct, can be useful.

```

2340 \def\exemplairepair{\ifodd\AMCid@etud}

```

**\AMClabel** Commands **\AMClabel**, **\AMCref** and **\AMCpageref** replaces L<sup>A</sup>T<sub>E</sub>X's **\label**, **\ref** and **\pageref** to be able to use different labels for different sheets.

```

\AMCref 2341 \newcommand\AMCstudentlabel[1]{\the\AMCid@etud-#1}
2342 \def\AMClabel#1{\expandafter\label{\AMCstudentlabel{#1}}}
2343 \def\AMCref#1{\expandafter\ref{\AMCstudentlabel{#1}}}
2344 \def\AMCpageref#1{\expandafter\pageref{\AMCstudentlabel{#1}}}

```

**\AMCqlabel** A label can be created for current question with **\AMCqlabel{<label>}**. This label can be used with **\AMCref** and **\AMCpageref**. This command is defined for backward compatibility only, since **\AMClabel** can also be used.

```

2345 \newcommand{\AMCqlabel}[1]{%
2346     \AMClabel{#1}%
2347 }

```

## 4.18 Pre-association

**\AMCassociation** Association between sheets and students can be made before the exam with the **\AMCassociation[<filename>]{<id>}** command. The optional argument *<filename>* will be used when printing student sheets to files.

```

2348 \newcommand{\AMCassociation}[2][{}]{%
2349     \ifAMC@calibration%
2350     \immediate\write\AMC@XYFILE{\string\association{\the\AMCid@etud}{#2}{#1}}%
2351     \fi%
2352 }

```

**\AMCstudentslistfile** You can also pass AMC the path to the CSV file with students, and the unique key that can be used, with **\AMCstudentslistfile{<path>}{<key>}**.

```

2353 \newcommand{\AMCstudentslistfile}[2]{%
2354     \ifAMC@calibration%
2355     \immediate\write\AMC@XYFILE{\string\with{studentslistfile=#1}}%

```

```

2356 \immediate\write\AMC@XYFILE{\string\with{studentslistkey=#2}}}%
2357 \fi%
2358 }

```

## 4.19 Package options

See section 3.1 for the options descriptions.

```

2359 \def\AMC@lang@code{}
2360 \DeclareOptionX{noshuffle}{\AMC@ordretrue}
2361 \DeclareOptionX{noshufflegroups}{\AMC@shuffleGfalse}
2362 \DeclareOptionX{fullgroups}{\AMC@fullGroupstrue}
2363 \DeclareOptionX{answers}{\AMC@corretheadtrue\AMC@correcttrue}
2364 \DeclareOptionX{indivanswers}{\AMC@correcttrue}
2365 \DeclareOptionX{textpos}{\AMC@textPostrue}
2366 \DeclareOptionX{extractonly}{\AMC@extractOnlytrue\AMC@textPostrue\AMC@boxStyle{shape=none}\AMC@boxedAnswers}
2367 \DeclareOptionX{box}{\AMC@qbloctrue}
2368 \DeclareOptionX{asbox}{\AMC@asqbloctrue}
2369 \DeclareOptionX{separateanswerssheet}{\AMC@ensembletrue}
2370 \DeclareOptionX{digits}{\AMC@inside@digittrue}
2371 \DeclareOptionX{ordre}{\AMC@ordretrue}
2372 \DeclareOptionX{correc}{\AMC@corretheadtrue\AMC@correcttrue}
2373 \DeclareOptionX{modele}{\AMC@corretheadtrue\AMC@correcfalse\AMC@ordretrue}
2374 \DeclareOptionX{correcindiv}{\AMC@correcttrue}
2375 \DeclareOptionX{init}{\AMC@SR@time}
2376 \DeclareOptionX{bloc}{\AMC@qbloctrue}
2377 \DeclareOptionX{completemulti}{\AMC@complete@multitrue}
2378 \DeclareOptionX{insidebox}{\AMC@inside@boxtrue}
2379 \DeclareOptionX{ensemble}{\AMC@ensembletrue}
2380 \DeclareOptionX{chiffres}{\AMC@inside@digittrue}
2381 \DeclareOptionX{outsidebox}{\AMC@outside@boxtrue}
2382 \DeclareOptionX{calibration}{\AMC@calibrationtrue}
2383 \DeclareOptionX{nowatermark}{\AMC@watermarkfalse}
2384 \newcommand\AMC@catalogMode{%
2385   \AMC@catalogtrue%
2386   \AMC@watermarkfalse\AMC@corretheadtrue%
2387   \AMC@correcttrue\AMC@ordretrue\AMC@shuffleGfalse%
2388   \AMC@fullGroupstrue%
2389   \def\AMC@intituleHead{\AMC@loc@catalog}\AMC@affichekeystrue}
2390 \newcommand\AMC@keys@next{\AMC@keys@linefalse}
2391 \newcommand\AMC@keys@line{\AMC@keys@linetrue}
2392 \DeclareOptionX{catalog}{\AMC@catalogMode}
2393 \DeclareOptionX{keys}[next]{\csname AMC@keys@#1\endcsname{}}
2394 \DeclareOptionX{francais}{\def\AMC@lang@code{FR}\AMC@loc@FR}
2395 \DeclareOptionX{lang}{\def\AMC@lang@code{#1}\csname AMC@loc@#1\endcsname}
2396 \DeclareOptionX{versionA}{%
2397   \def\AMC@id@checkmax{31}\def\AMC@NCBetud{9}\def\AMC@NCBpage{4}%
2398   \def\AMC@NCBcheck{5}\setlength{\AMC@CBtaille}{4cm}%
2399   \def\AMC@premierecopie{100}}
2400 \DeclareOptionX{plain}{\AMC@plaintrue}
2401 \DeclareOptionX{nopage}{\AMC@pagelayoutfalse}
2402 \DeclareOptionX{postcorrect}{\AMC@postcorrecttrue}
2403 \DeclareOptionX{automarks}{\AMC@automarkstrue}
2404 \newif\ifAMCneeds@storebox\AMCneeds@storeboxfalse

```

```

2405 \DeclareOptionX{storebox}{\AMCneeds@storeboxtrue}
2406 \DeclareOptionX{pdfform}{\AMC@pdfformtrue}
2407 \DeclareOptionX{codedigit}{\AMCcodeID@{#1}}
2408 \newif\ifAMC@survey\AMC@surveyfalse
2409 \DeclareOptionX{survey}{\AMC@surveytrue}
2410 \ProcessOptionsX
2411 \ifAMCneeds@storebox
2412   \RequirePackage{storebox}\AtBeginDocument{{}}%
2413 \fi
2414 \ifAMC@pdfform
2415   \AMCmessage{VAR:project:pdfform=1}%
2416   \AMCboxStyle{shape=form}%
2417   \RequirePackage[pageanchor=false]{hyperref}%
2418 \else%
2419   \AMCmessage{VAR:project:pdfform=0}%
2420 \fi
2421 \AtBeginDocument{%
2422   \ifAMCneeds@storebox%
2423     \let\AMC@new@savebox=\newstorebox%
2424     \let\AMC@save@box=\storebox%
2425     \let\AMC@use@box=\usestorebox%
2426   \fi%
2427   \AMC@new@savebox{\AMC@ovalbox@R}%
2428   \AMC@new@savebox{\AMC@ovalbox@RF}%
2429   \AMC@new@savebox{\AMC@ovalbox@}%
2430   \AMC@new@savebox{\AMC@ovalbox@F}%
2431   \AMC@shapeprepare%
2432 }

```

## 4.20 Survey add-on

Some code and *tikz* settings to help handling surveys, see [https://survey.codes/pdf/surveyamc\\_manual.pdf](https://survey.codes/pdf/surveyamc_manual.pdf) for more details. This survey add-on is originally written by Claudia Saalbach.

### Questionnaires

```

auto 2433 \ifAMC@survey
question-auto 2434 \NewEnviron{Questionnaires}[1]{
  values 2435 \onecopy{#1}{
    values-auto 2436 \BODY
  variable-auto 2437 }
  \answer 2438 }
  2439 \NewEnviron{auto}[1]{
    2440 \csvreader[head to column names, separator=tab]{#1}{}{
    2441 \BODY
    2442 }
    2443 }
    2444 \NewEnviron{question-auto}[3]{
    2445 \csvreader[head to column names, separator=tab]{#1}{}{
    2446 \ifcsvstrcmp{#2}{#3}{\BODY \\\}{}
    2447 }
    2448 }
    2449 \newenvironment{values}{}{}

```

```

2450 \NewEnviron{values-auto}[5]{
2451 \csvreader[head to column names, separator=tab]{#1}{}{
2452 \ifcsvstrcmp{#2}{#3}{
2453 \ifcsvstrcmp{#4}{#5}{\BODY \\\
2454 }{}
2455 }{}
2456 }
2457 }
2458 \NewEnviron{variable-auto}[3]{
2459 \foreach \x in {#3}{
2460 \csvreader[head to column names, separator=tab]{#1}{}{
2461 \ifcsvstrcmp{#2}{\x}{\BODY}{}
2462 }
2463 }
2464 }
2465 \newcommand{\answer}[5][\global\advance\AMCrep@count\@ne\relax%
2466 \ifAMC@calibration\AMCmessage{REP=\the\AMCrep@count:B}\fi%
2467 \global\AMCune@bonnettrue%
2468 \AMCload@reponse{\une@rep{\ifAMC@correc\AMC@box{#1}{\AMC@checkbox}%
2469 \else\AMC@box{#1}{\fi}{#2}{#3}{#4}{#5}}{\the\AMCrep@count}\ignorespaces}
2470 \RequirePackage{tikz}
2471 \usetikzlibrary{positioning, shapes, arrows, tikzmark, decorations.pathreplacing}
2472 \tikzset{
2473 checkbox-sc/.style={
2474 right=of lab\thecsvrow
2475 },
2476 vallab-sc/.style={
2477 text width=4cm,
2478 align=left,
2479 },
2480 checkbox-mc/.style={
2481 },
2482 vallab-mc/.style={
2483 above=of box\thecsvrow,
2484 text width=1.4cm,
2485 align=center,
2486 },
2487 varlab-mc/.style={
2488 text width=4cm,
2489 align=left,
2490 },
2491 node distance= 0mm,
2492 }
2493 \fi

```

## 4.21 Package Errors

**AMC@error@explain** Error to display if \explain command is used outside question like environments

```

2494 \def\AMC@error@explain{\PackageError{automultiplechoice}{
2495 Command \protect\explain\space can only be used inside\MessageBreak question like environments}{Some
2496 }}

```

## 4.22 Optional features

This package tries to see if optional packages `environ` and `etex` are loadable, and load them if possible. This behaviour can be cancelled by using `plain` option.

```
2497 \ifAMC@plain
2498 \else
2499   \IfFileExists{environ.sty}{\RequirePackage{environ}}{}
2500   \ifx\TeXversion\@undefined
2501     \else
2502       \RequirePackage{etex}
2503     \fi
2504 \fi
```

`examcopy` Then, if `environ` package is loaded and defines command `\NewEnviron`, environment `examcopy` is defined.

Environment `{examcopy}[\langle n \rangle]` does the same as command `onecopy`: it encloses  $\text{\LaTeX}$  code which makes *one* exam copy. Optional argument  $\langle n \rangle$  gives the number of desired copies – this can also be modified redefinig `\AMCNombreCopies`.

```
2505 \@ifpackageloaded{environ}{%
2506   \ifx\NewEnviron\undefined\PackageWarning{automultiplechoice}%
2507     {Package environ loaded but too old version:
2508      environnement examcopy/copieexamen will NOT be defined.}%
2509   \else\NewEnviron{examcopy}[1][5]{\onecopy{#1}{\BODY}}\fi}%
2510 {\PackageWarning{automultiplechoice}%
2511   {Package environ not loaded: environnement
2512    examcopy/copieexamen will NOT be defined.}}
```

## 4.23 Use with recent LuaTeX versions

In recent LuaTeX versions, the commands `pdfsavepos`, `pdflastxpos` and `pdflastypos` has been renamed, stripping the `pdf` part. The following code tries to detect this situation and make the bindings between the old and new command names.

```
2513 \ExplSyntaxOn
2514
2515 \cs_if_exist:NTF \pdfsavepos { } {
2516   \cs_if_exist:NTF \savepos { \cs_new_eq:NN \pdfsavepos \savepos } { }
2517 }
2518 \cs_if_exist:NTF \pdflastxpos { } {
2519   \cs_if_exist:NTF \lastxpos { \cs_new_eq:NN \pdflastxpos \lastxpos } { }
2520 }
2521 \cs_if_exist:NTF \pdflastypos { } {
2522   \cs_if_exist:NTF \lastypos { \cs_new_eq:NN \pdflastypos \lastypos } { }
2523 }
```

In some situations, the *page* dimensions are different from the *paper* dimensions. This must be taken into account when computing coordinates.

```
2524
2525 \cs_if_exist:NTF \pdfpagewidth { } {
2526   \cs_new_eq:NN \pdfpagewidth \paperwidth
2527 }
2528 \cs_if_exist:NTF \pdfpageheight { } {
2529   \cs_new_eq:NN \pdfpageheight \paperheight
```

```

2530 }
2531
2532 \ExplSyntaxOff

```

## 4.24 External control

Some of the package options can be controlled defining `\xxxExterne` commands. For example, the following command will format the subject document, whatever options are used in the  $\text{\LaTeX}$  file:

```

\pdflatex '\nonstopmode\def\SujetExterne{1}\def\NoWatermarkExterne{1}\input{mcq.tex}'

```

```

2533 \ifx\SujetExterne\undefined\else
2534 \message{***SUJET***^~J}
2535 \AMC@calibrationtrue\AMC@correcfalse\AMC@correcheadfalse\AMC@watermarkfalse
2536 \fi
2537 \ifx\ScoringExterne\undefined\else
2538 \message{***SCORING***^~J}
2539 \AMC@calibrationtrue\AMC@correcfalse\AMC@correcheadfalse\AMC@watermarkfalse\AMC@invisibletrue
2540 \fi
2541 \ifx\CorrigeExterne\undefined\else
2542 \message{***CORRIGE***^~J}
2543 \AMC@calibrationfalse\AMC@correcheadtrue\AMC@correcttrue\AMC@watermarkfalse
2544 \fi
2545 \ifx\CorrigeIndivExterne\undefined\else
2546 \message{***CORRIGE***^~J}
2547 \AMC@calibrationfalse\AMC@correcheadfalse\AMC@correcttrue\AMC@watermarkfalse
2548 \fi
2549 \ifx\CatalogExterne\undefined\else
2550 \message{***CATALOG***^~J}
2551 \AMC@catalogMode
2552 \fi
2553 \ifx\NoWatermarkExterne\undefined\else
2554 \AMC@watermarkfalse
2555 \fi
2556 \ifx\codeDigitExterne\undefined\else
2557 \AMCcodeID@{\codeDigitExterne}
2558 \fi

```

## 4.25 Page layout

The following code sets the correct page layout to have room for signs for scan analysis, and prepares watermark printing:

```

2559 \@ifpackageloaded{geometry}{\usepackage{geometry}}
2560 \ifAMC@pagelayout
2561 \ifAMC@correchead
2562 \geometry{hmargin=3cm,vmargin={1cm,1cm},includeheadfoot,headheight=1cm,footskip=1cm}
2563 \else
2564 \geometry{hmargin=3cm,headheight=2cm,headsep=.3cm,footskip=1cm,top=3.5cm,bottom=2.5cm}
2565 \fi
2566 \ifAMC@watermark
2567 \ifAMC@correchead\else
2568 \def\AMC@note{\begin{minipage}{0.65\linewidth}

```



```

2569      \AMC@LR{\textcolor{blue}{\AMC@loc@message}}
2570      \end{minipage}
2571    }
2572  \fi
2573 \fi
2574 \fi

```

## 4.26 Initialisation

Initialisation of the check counter:

```
2575 \AMCid@check=\AMCid@checkmax\advance\AMCid@check\@ne
```

Telling outside if separate answer sheet, and boxes labelling, are requested:

```

2576 \ifAMC@ensemble\AMCmessage{VAR:ensemble=1}\fi
2577 \ifAMC@inside@box\AMCmessage{VAR:insidebox=1}\fi
2578 \ifAMC@outside@box\AMCmessage{VAR:outsidebox=1}\fi
2579 \ifAMC@postcorrect\AMCmessage{VAR:postcorrect=1}\fi

```

Preparing writing to .xy file :

```

2580 \ifAMC@calibration
2581 \newwrite\AMC@XYFILE%
2582 \immediate\openout\AMC@XYFILE\jobname.xy%
2583 \immediate\write\AMC@XYFILE{\string\version{\AMC@VERSION}}
2584 \immediate\write\AMC@XYFILE{\string\with{codedigit=\AMCcodeID@mode}}
2585 \immediate\write\AMC@XYFILE{\string\with{version=\AMC@VERSION}}
2586 \immediate\write\AMC@XYFILE{\string\with{ensemble=\ifAMC@ensemble yes\else no\fi}}
2587 \immediate\write\AMC@XYFILE{\string\with{insidebox=\ifAMC@inside@box yes\else no\fi}}
2588 \immediate\write\AMC@XYFILE{\string\with{outsidebox=\ifAMC@outside@box yes\else no\fi}}
2589 \immediate\write\AMC@XYFILE{\string\with{postcorrect=\ifAMC@postcorrect yes\else no\fi}}
2590 \immediate\write\AMC@XYFILE{\string\with{extractonly=\ifAMC@extractOnly yes\else no\fi}}
2591 \immediate\write\AMC@XYFILE{\string\with{lang=\AMC@lang@code}}
2592 \ifx\AMCNombreCopies\undefined%
2593 \immediate\write\AMC@XYFILE{\string\with{ncopies=default}}%
2594 \else%
2595 \immediate\write\AMC@XYFILE{\string\with{ncopies=\AMCNombreCopies}}%
2596 \fi%
2597 \fi

```

## 4.27 French command names

For backward compatibility, a lot of commands have their french counterpart:

```

2598 \let\reponses=\choices\let\endreponses=\endchoices
2599 \let\reponseshoriz=\choiceshoriz\let\endreponseshoriz=\endchoiceshoriz
2600 \let\reponsesperso=\choicescustom\let\endreponsesperso=\endchoicescustom
2601 \let\bonne=\correctchoice
2602 \let\mauvaise=\wrongchoice
2603 \let\bareme=\scoring
2604 \let\baremeDefautM=\scoringDefaultM
2605 \let\baremeDefautS=\scoringDefaultS
2606 \def\exemplaire{\AMC@loc@FR\onecopy}
2607 \@ifpackageloaded{environ}{%
2608   \let\copieexamen=\examcopy\let\endcopieexamen=\endexamcopy}{%
2609 \let\melangegroupe=\shufflegroup

```

```

2610 \let\restituegroupe=\insertgroup
2611 \let\alafin=\lastchoices
2612 \let\formulaire=\AMCform
2613 \let\AMCdebutFormulaire=\AMCformBegin
2614 \let\champnom=\namefield
2615 \let\choixIntervalles=\AMCIntervals

```

## 5 Outputs

In the .xy file, 0/⟨*n*⟩ means student sheet number 0 (there is only one “student sheet” numbered 0 for this document as we did not use \onecopy) and page number ⟨*n*⟩ inside this student sheet. Then, each instance of the \tracepos command shows *x* and *y* positions as arguments #2 and #3 (unit is sp, such that 65536 × 72.27 sp is one inch). One has to take min and max of the *x*-values to determine the left and right position of the box, and min and max values of *y*-values to determine top and bottom position of the box.

### 5.1 namefield command

Lines in the .xy file from a \namefield command:

```

\tracepos{0/35: __zone: id: __n}{0sp}{48735204sp}{square}
\tracepos{0/35: __zone: id: __n}{5873801sp}{0sp}{square}
\tracepos{0/35: __zone: id: __n}{15861297sp}{0sp}{square}
\tracepos{0/35: __zone: id: __n}{0sp}{45750026sp}{square}

```

### 5.2 AMCboxedchar command

Lines in the .xy file from a \AMCboxedchar command:

```

\tracepos{0/35: test}{22855914sp}{32615979sp}{square}
\tracepos{0/35: test}{23561334sp}{31910559sp}{square}

```

### 5.3 AMCcode command

Lines in the .xy file from a \AMCcode command. Here, code[⟨*n*⟩]:⟨*q*⟩,⟨*v*⟩ relates to digit number ⟨*n*⟩ from the right (⟨*n*⟩=1 for units, ⟨*n*⟩=2 for tens, ⟨*n*⟩=3 for hundreds and so on), question number ⟨*q*⟩ (\AMCcode uses a fake question; this number can be ignored), and value ⟨*v*⟩-1 (box number ⟨*v*⟩ for the digit).

```

\tracepos{0/58: case: code[5]: 16,1}{24875504sp}{38484742sp}{square}
\tracepos{0/58: case: code[5]: 16,1}{25580924sp}{37779322sp}{square}
\boxchar{0/58: case: code[5]: 16,1}{A}
\tracepos{0/58: case: code[5]: 16,2}{24875504sp}{37370630sp}{square}
\tracepos{0/58: case: code[5]: 16,2}{25580924sp}{36665210sp}{square}
\boxchar{0/58: case: code[5]: 16,2}{B}
\tracepos{0/58: case: code[5]: 16,3}{24875504sp}{36256518sp}{square}
\tracepos{0/58: case: code[5]: 16,3}{25580924sp}{35551098sp}{square}
\boxchar{0/58: case: code[5]: 16,3}{C}
\tracepos{0/58: case: code[5]: 16,4}{24875504sp}{35142406sp}{square}
\tracepos{0/58: case: code[5]: 16,4}{25580924sp}{34436986sp}{square}
\boxchar{0/58: case: code[5]: 16,4}{D}

```

```

\tracepos{0/58:case:code[4]:17,1}{26540303sp}{40712966sp}{square}
\tracepos{0/58:case:code[4]:17,1}{27245723sp}{40007546sp}{square}
\boxchar{0/58:case:code[4]:17,1}{0}
\tracepos{0/58:case:code[4]:17,2}{26540303sp}{39598854sp}{square}
\tracepos{0/58:case:code[4]:17,2}{27245723sp}{38893434sp}{square}
\boxchar{0/58:case:code[4]:17,2}{1}
\tracepos{0/58:case:code[4]:17,3}{26540303sp}{38484742sp}{square}
\tracepos{0/58:case:code[4]:17,3}{27245723sp}{37779322sp}{square}
\boxchar{0/58:case:code[4]:17,3}{2}
\tracepos{0/58:case:code[4]:17,4}{26540303sp}{37370630sp}{square}
\tracepos{0/58:case:code[4]:17,4}{27245723sp}{36665210sp}{square}
\boxchar{0/58:case:code[4]:17,4}{3}
\tracepos{0/58:case:code[4]:17,5}{26540303sp}{36256518sp}{square}
\tracepos{0/58:case:code[4]:17,5}{27245723sp}{35551098sp}{square}
\boxchar{0/58:case:code[4]:17,5}{4}
\tracepos{0/58:case:code[4]:17,6}{26540303sp}{35142406sp}{square}
\tracepos{0/58:case:code[4]:17,6}{27245723sp}{34436986sp}{square}
\boxchar{0/58:case:code[4]:17,6}{5}
\tracepos{0/58:case:code[3]:18,1}{28032160sp}{40712966sp}{square}
\tracepos{0/58:case:code[3]:18,1}{28737580sp}{40007546sp}{square}
\boxchar{0/58:case:code[3]:18,1}{0}
\tracepos{0/58:case:code[3]:18,2}{28032160sp}{39598854sp}{square}
\tracepos{0/58:case:code[3]:18,2}{28737580sp}{38893434sp}{square}
\boxchar{0/58:case:code[3]:18,2}{1}
\tracepos{0/58:case:code[3]:18,3}{28032160sp}{38484742sp}{square}
\tracepos{0/58:case:code[3]:18,3}{28737580sp}{37779322sp}{square}
\boxchar{0/58:case:code[3]:18,3}{2}
\tracepos{0/58:case:code[3]:18,4}{28032160sp}{37370630sp}{square}
\tracepos{0/58:case:code[3]:18,4}{28737580sp}{36665210sp}{square}
\boxchar{0/58:case:code[3]:18,4}{3}
\tracepos{0/58:case:code[3]:18,5}{28032160sp}{36256518sp}{square}
\tracepos{0/58:case:code[3]:18,5}{28737580sp}{35551098sp}{square}
\boxchar{0/58:case:code[3]:18,5}{4}
\tracepos{0/58:case:code[3]:18,6}{28032160sp}{35142406sp}{square}
\tracepos{0/58:case:code[3]:18,6}{28737580sp}{34436986sp}{square}
\boxchar{0/58:case:code[3]:18,6}{5}
\tracepos{0/58:case:code[2]:19,1}{29524017sp}{40712966sp}{square}
\tracepos{0/58:case:code[2]:19,1}{30229437sp}{40007546sp}{square}
\boxchar{0/58:case:code[2]:19,1}{0}
\tracepos{0/58:case:code[2]:19,2}{29524017sp}{39598854sp}{square}
\tracepos{0/58:case:code[2]:19,2}{30229437sp}{38893434sp}{square}
\boxchar{0/58:case:code[2]:19,2}{1}
\tracepos{0/58:case:code[2]:19,3}{29524017sp}{38484742sp}{square}
\tracepos{0/58:case:code[2]:19,3}{30229437sp}{37779322sp}{square}
\boxchar{0/58:case:code[2]:19,3}{2}
\tracepos{0/58:case:code[2]:19,4}{29524017sp}{37370630sp}{square}
\tracepos{0/58:case:code[2]:19,4}{30229437sp}{36665210sp}{square}
\boxchar{0/58:case:code[2]:19,4}{3}
\tracepos{0/58:case:code[2]:19,5}{29524017sp}{36256518sp}{square}

```

```

\tracepos{0/58:case:code[2]:19,5}{30229437sp}{35551098sp}{square}
\boxchar{0/58:case:code[2]:19,5}{4}
\tracepos{0/58:case:code[2]:19,6}{29524017sp}{35142406sp}{square}
\tracepos{0/58:case:code[2]:19,6}{30229437sp}{34436986sp}{square}
\boxchar{0/58:case:code[2]:19,6}{5}
\tracepos{0/58:case:code[1]:20,1}{31015874sp}{40712966sp}{square}
\tracepos{0/58:case:code[1]:20,1}{31721294sp}{40007546sp}{square}
\boxchar{0/58:case:code[1]:20,1}{0}
\tracepos{0/58:case:code[1]:20,2}{31015874sp}{39598854sp}{square}
\tracepos{0/58:case:code[1]:20,2}{31721294sp}{38893434sp}{square}
\boxchar{0/58:case:code[1]:20,2}{1}
\tracepos{0/58:case:code[1]:20,3}{31015874sp}{38484742sp}{square}
\tracepos{0/58:case:code[1]:20,3}{31721294sp}{37779322sp}{square}
\boxchar{0/58:case:code[1]:20,3}{2}
\tracepos{0/58:case:code[1]:20,4}{31015874sp}{37370630sp}{square}
\tracepos{0/58:case:code[1]:20,4}{31721294sp}{36665210sp}{square}
\boxchar{0/58:case:code[1]:20,4}{3}
\tracepos{0/58:case:code[1]:20,5}{31015874sp}{36256518sp}{square}
\tracepos{0/58:case:code[1]:20,5}{31721294sp}{35551098sp}{square}
\boxchar{0/58:case:code[1]:20,5}{4}
\tracepos{0/58:case:code[1]:20,6}{31015874sp}{35142406sp}{square}
\tracepos{0/58:case:code[1]:20,6}{31721294sp}{34436986sp}{square}
\boxchar{0/58:case:code[1]:20,6}{5}

```

# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
<b>2</b>	<b>Samples</b>	<b>1</b>
2.1	Standard layout . . . . .	4
2.2	Separate answer sheet . . . . .	5
2.3	Without markers . . . . .	6
<b>3</b>	<b>Usage</b>	<b>10</b>
3.1	Package options . . . . .	10
3.2	Questions and answers . . . . .	11
3.3	Scoring . . . . .	13
3.4	Groups of questions . . . . .	14
3.5	Students identification . . . . .	15
3.6	Separate answer sheet . . . . .	16
3.7	Random computation questions . . . . .	17
3.8	French command names . . . . .	20
3.9	Customisation . . . . .	20
3.9.1	Boxes . . . . .	20
3.9.2	Codes . . . . .	22
3.9.3	Answers . . . . .	22
<b>4</b>	<b>Implementation</b>	<b>22</b>
4.1	Variables . . . . .	23
4.2	Dimensions . . . . .	25
4.3	Human readable sheet ID position . . . . .	26
4.4	Localisation . . . . .	27
4.4.1	English . . . . .	27
4.4.2	Catalan . . . . .	27
4.4.3	Dutch . . . . .	28
4.4.4	French . . . . .	28
4.4.5	German . . . . .	28
4.4.6	Italian . . . . .	29
4.4.7	Norwegian . . . . .	29
4.4.8	Portuguese . . . . .	29
4.4.9	Spanish . . . . .	30
4.4.10	Japanese . . . . .	30
4.4.11	Other languages . . . . .	30
4.5	Interaction with other packages . . . . .	30
4.5.1	cleveref . . . . .	30
4.6	Random . . . . .	31
4.6.1	Random pseudo-generator . . . . .	31
4.6.2	Uniform random deviates . . . . .	31
4.6.3	Tokens shuffling . . . . .	31
4.7	Keys numbering . . . . .	32
4.8	Boxes . . . . .	32
4.8.1	Character logging . . . . .	32
4.8.2	Position logging . . . . .	33
4.8.3	Boxes to be checked by students . . . . .	35

4.8.4	Scoring zones . . . . .	40
4.8.5	Binary boxes . . . . .	40
4.9	Checking Environment . . . . .	42
4.10	Handling groups of questions . . . . .	42
4.11	Questions . . . . .	46
4.11.1	Managing answers . . . . .	47
4.11.2	Separate answer sheet . . . . .	48
4.11.3	Formatting answers . . . . .	51
4.11.4	Score zones . . . . .	53
4.11.5	Formatting questions . . . . .	55
4.11.6	Explanations . . . . .	57
4.12	Scoring . . . . .	57
4.13	Numerical data . . . . .	58
4.13.1	Codes . . . . .	58
4.13.2	Numerical questions . . . . .	61
4.13.3	Intervals . . . . .	75
4.14	Open questions . . . . .	76
4.15	Boxes with letters only . . . . .	79
4.16	Page formatting . . . . .	80
4.16.1	Watermark . . . . .	80
4.16.2	Signs for scan analysis . . . . .	80
4.17	Defining a single exam copy content . . . . .	82
4.18	Pre-association . . . . .	83
4.19	Package options . . . . .	84
4.20	Survey add-on . . . . .	85
4.21	Package Errors . . . . .	86
4.22	Optional features . . . . .	87
4.23	Use with recent LuaTeX versions . . . . .	87
4.24	External control . . . . .	88
4.25	Page layout . . . . .	88
4.26	Initialisation . . . . .	89
4.27	French command names . . . . .	89
<b>5</b>	<b>Outputs</b>	<b>90</b>
5.1	<code>namefield</code> command . . . . .	90
5.2	<code>AMCboxedchar</code> command . . . . .	90
5.3	<code>AMCcode</code> command . . . . .	90

Numbers written in *italic* refer to the page where the corresponding entry is described; numbers underlined refer to the code line of the definition; numbers in **roman** refer to the code lines where the entry is used.

95

\AMC@error@explain .....	1243, 1245, <u>2494</u>	\AMC@loc@none ..	100, 113, 126, 140, 154, 167, 180, 193, 206, 219, 838	\AMC@new@savebox .....	.. 359, 2423, 2427–2430
\AMC@fillcolor@ .....	.. 379, 397, 398, 414, 419	\AMC@loc@PT .....	185	\AMC@note .....	2233, 2234, 2268, 2278, 2568
\AMC@fin@rep ..	<u>825</u> , 1010, 1014, 1018, 1021, 1024	\AMC@loc@q .....	96, 109, 123, 136, 150, 164, 177, 189, 203, 215, 1174	\AMC@numeric@alsocorrect .....	1591, 1943, 1947, 1957, 1961
\AMC@formBox .....	<u>519</u>	\AMC@loc@qf .....	95, 108, 122, 135, 149, 163, 176, 188, 202, 214, 847	\AMC@numeric@keepas .....	1589, 1936, 1937
\AMC@formBox@ .....	<u>519</u>	\AMC@loc@question .....	101, 114, 127, 141, 155, 168, 181, 194, 207, 220, 226	\AMC@numeric@scoreapprox .....	1581, 1902, 1912
\AMC@fullGroupsfalse ...	34	\AMC@loc@questions .....	102, 115, 128, 142, 156, 169, 182, 195, 208, 221, 226	\AMC@numeric@scoreexact .....	1579, 1901, 1907
\AMC@fullGroupstrue .....	2362, 2388	\AMC@LR .....	<u>21</u> , 389, 2208, 2255–2259, 2569	\AMC@numeric@scorewrong .....	1583, 1909, 1912
\AMC@if@separate@question .....	876, 951, 956, 964, 970, 1402, 2017, 2163, 2199	\AMC@makeovalbox .....	412, 431–434	\AMC@numeration .....	278, 282, 283
\AMC@imax .....	712, 724–726	\AMC@mem@add .....	906, 952, 959, 966, 973, 1403	\AMC@outside@sep ..	494, 500
\AMC@intituleHead .....	2232, 2249, 2389	\AMC@mem@add@ifneeded ..	<u>863</u> , 1212	\AMC@oval@radius .....	419, 493, 510, 512
\AMC@keepmemoryfalse ..	2315	\AMC@mem@addsingle@ifneeded ..	955, 995, 997, 999, 1001	\AMC@ovalbox@ .....	433, 447, 2429
\AMC@keepmemorytrue ...	992	\AMC@mem@addvar .....	911	\AMC@ovalbox@F .....	434, 445, 2430
\AMC@keyBox@ .....	.. 548, 1295, 1614, 1617	\AMC@mem@aid .....	930, 965	\AMC@ovalbox@R .....	430, 431, 453, 2427
\AMC@keys@line .....	2391	\AMC@mem@answer .....	962, 1025, 1032–1034, 1039, 1045	\AMC@ovalbox@RF .....	432, 451, 2428
\AMC@keys@next .....	2390	\AMC@mem@category ..	924, 958, 2017, 2163, 2199	\AMC@pagepos .....	<u>292</u> , 2245
\AMC@lang@code .....	.. 2359, 2394, 2395, 2591	\AMC@mem@clear ...	881, 2317	\AMC@premierecopie .....	.. 600, 2301, 2304, 2399
\AMC@loc@CA .....	104	\AMC@mem@clear ..	901, 957, 971	\AMC@prepare ..	281, 284, 285
\AMC@loc@catalog ..	98, 111, 125, 138, 152, 166, 179, 191, 205, 217, 2389	\AMC@mem@next ..	969, 1211	\AMC@prepare@element ...	.. 675, 683, 798
\AMC@loc@corrected .....	97, 110, 124, 137, 151, 165, 178, 190, 204, 216, 2232	\AMC@mem@openQuestion ..	918, 972	\AMC@printformoutside@false .....	523
\AMC@loc@DE .....	145	\AMC@mn@leftmargin .....	1062, 1070, 1073, 1079	\AMC@printformoutside@true .....	525, 526
\AMC@loc@draft ..	92, 105, 119, 132, 146, 160, 173, 186, 199, 212, 2203	\AMC@mn@rightmargin .....	1063, 1071, 1074, 1077	\AMC@printkeyoutside@false .....	540
\AMC@loc@ES .....	198	\AMC@mn@sep ..	1061, 1077, 1079	\AMC@printkeyoutside@true .....	542, 544
\AMC@loc@explain ..	99, 112, 139, 153, 192, 218, 1243	\AMC@mn@test ..	1060, 1067, 1076	\AMC@qaff ..	848, <u>1164</u> , 1206, 1239
\AMC@loc@FR ..	131, 2394, 2606	\AMC@multi@report ..	1322, 2322	\AMC@save@box ..	360, 415, 2424
\AMC@loc@IT .....	159	\AMC@multiclear ..	1327, 2314	\AMC@setcolors@ .....	.. 375, 396, 413, 438
\AMC@loc@JA .....	211	\AMC@NCBcheck ..	<u>595</u> , 2238, 2398	\AMC@shape@form .....	472
\AMC@loc@message ..	93, 106, 120, 133, 147, 161, 174, 187, 200, 213, 2569	\AMC@NCBetud ..	<u>595</u> , 2236, 2397	\AMC@shape@form@base ..	459, 473
\AMC@loc@namesurname ...	.. 103, 116, 129, 143, 157, 170, 183, 196, 209	\AMC@NCBpage ..	<u>595</u> , 2237, 2397	\AMC@shape@form@ticked ..	.. 461, 463, 466
\AMC@loc@NL .....	118			\AMC@shape@none .....	476
\AMC@loc@NO .....	172			\AMC@shape@oval .....	437
				\AMC@shape@square .....	394
				\AMC@shapename .....	.. 292, 300, 309, 318, 495
				\AMC@shapename@ ..	292, 390, 506



\AMC@shapeprepare .....	\AMC@sza@box 1118, 1126, 1129	\AMCbeginAnswer .....
..... 506, 514, 2431	\AMC@sza@callin 857, 858,	. 1017, <u>1046</u> , 2115, 2177
\AMC@shapeprepare@form 458	1140, 1145, 1153, 1155	\AMCbeginQuestion .....
\AMC@shapeprepare@none 475	\AMC@sza@callin@margin 1130	..... <u>1173</u> , 1206,
\AMC@shapeprepare@oval 429	\AMC@sza@callin@margins 1134	1219, 1225, 1239, 1278
\AMC@shapeprepare@square 393	\AMC@sza@callin@none .. 1122	\AMCbin@begin .....
\AMC@shuffletoks .....	\AMC@sza@callin@question	650
..... 259, 745, 748, 818	..... 1126	\AMCbin@did 605, 613, 616,
\AMC@smashbox .....	\AMC@sza@callout 853, 854,	618, 623, 626, 628, 634
..... 363, 366–370, 373	1139, 1144, 1148, 1150	\AMCbin@digit .....
\AMC@smashboxheight .....	\AMC@sza@callout@margin 1129	. 603, 636, 639, 645, 648
..... 364, 367–369, 372	\AMC@sza@callout@margins	\AMCbin@hsep 607, 610, 614, 624
\AMC@smashcentered .....	..... 1133	\AMCbin@id .....
. 365, 407, 408, 440, 441	\AMC@sza@callout@none . 1121	604,
\AMC@SR ... 228, 230, 233–	\AMC@sza@callout@question	609, 617, 618, 627, 628
236, 238, 240, 241, 2308	..... 1125	\AMCbin@ndigits .....
\AMC@SR@count .....	\AMC@sza@depth .....	.... 606, 608, 645, 648
..... 233, 234, 236,	..... 1118, 1138, 1143	\AMCbin@number .....
244, 248, 251, 254–257	\AMC@sza@height .....	. 602, 635, 638, 640, 642
\AMC@SR@time .....	..... 1118, 1137, 1142	\AMCbin@one .....
245, 2375	\AMC@sza@init@margin .. 1128	612, 640
\AMC@SR@advance 231, 238, 239	\AMC@sza@init@margins . 1132	\AMCbin@sequence .. 601,
\AMC@SR@bit .....	\AMC@sza@init@none .... 1120	635, 640, 641, 646, 647
238	\AMC@sza@init@question 1124	\AMCbin@style .....
\AMC@SR@const .. 229, 233, 236	\AMC@sza@width .....	611, 635
\AMC@SR@max .....	..... 1118, 1136, 1141	\AMCbin@zero .. 622, 641, 646
<u>244</u> , 272	\AMC@tempenv .....	\AMCbloc .....
\AMC@SR@nextByte .....	656, 657	1180
<u>244</u>	\AMC@tracebox .....	\AMCbo@help . 2167, 2168, 2181
\AMC@SR@num 246, 247, 249,	. <u>292</u> , 346, 439, 465,	\AMCbo@hide .....
250, 254, 257, 272, 273	583, 585, 588, 1041, 2226	2194, 2200
\AMC@SR@set 230, 242, 243, 245	\AMC@tracechar <u>334</u> , 536, 554	\AMCbo@opts .....
\AMC@SR@test .....	\AMC@tracepos .....	2171
239, 250	..... 293, 404, 410,	\AMCbo@show .....
\AMC@SR@value .....	1027, 1030, 1204, 1212	2172, 2200
241	\AMC@traceposx .... 302, 322	\AMCbotext@goto .. 2166, 2195
\AMC@step@question .....	\AMC@traceposy 311, 321, 323	\AMCbox@color .....
..... 1164, 1198, 1236	\AMC@unnumero .....	517, 2260
\AMC@sti 259, 267, 271, 274, 275	<u>278</u>	\AMCbox@dimensions . 518, 635
\AMC@stil .....	\AMC@use@box .....	\AMCboxed@answers <u>1004</u> , 2366
. 260, 268–270, 272, 276	361,	\AMCboxed@only .....
\AMC@survey@false .....	445, 447, 451, 453, 2425	<u>2166</u>
2408	\AMC@VERSION .... 2583, 2585	\AMCbox@outside@letter ...
\AMC@survey@true .....	\AMC@watertext .. 2203, 2214	..... <u>519</u> , 2117, 2179
2409	\AMC@XYFILE .....	\AMCbox@style .....
\AMC@sz@box .....	295,	..... <u>20</u> , <u>488</u> , 2366, 2416
1083,	304, 313, 325, 330–	\AMCc@circle .....
1085, 1087, 1090, 1133	332, 336, 342, 344,	2222, 2225
\AMC@sz@callin .....	1185, 1319, 2307,	\AMCchoice@label ... <u>477</u> , 558
.. 1095, 1100, 1108,	2350, 2355, 2356,	\AMCchoice@label@format ..
1110, 1175, 1207, 1208	2581–2591, 2593, 2595	..... 391, <u>477</u>
\AMC@sz@callin@question 1085	\AMC@add@pagesto .....	\AMCc@clear@double@page <u>16</u> , 2333
\AMC@sz@callout .....	<u>2324</u>	\AMCc@code .....
..... 1094, 1099,	\AMC@answer .....	1396
1103, 1105, 1201, 1202	. 1045, <u>1046</u> , 2116, 2178	\AMCc@code@grid .....
\AMC@sz@callout@margin 1087	\AMC@association .....	<u>15</u> , <u>1257</u>
\AMC@sz@callout@margins 1090	<u>2348</u>	\AMCc@code@grid@int ... <u>15</u> , <u>1257</u>
\AMC@sz@depth 1083, 1093, 1098	\AMC@before@question .....	\AMCc@code@h .....
\AMC@sz@height .....	..... <u>1173</u> , 1200, 1279	1397
..... 1083, 1092, 1097		\AMCc@code@h@space .....
\AMC@sz@init@margins .. 1089		..... 1258, 1299, 1358
\AMC@sz@width 1083, 1091, 1096		\AMCc@code@id .....
		1263, 1338
		\AMCc@code@id@@ .....
		. 1262, 1266, 2407, 2557
		\AMCc@code@id@dot .....
		1261
		\AMCc@code@id@mode . 1264, 2584

\AMCcodeID@squarebrackets	\AMCid@check	\AMCload@@reponse
..... 1260	27, 327, 2238, 2243,	831, 1052, 1056, 2468
\AMCcodeVspace	2244, 2261, 2265, 2575	\AMCload@counter
. 1259, 1297, 1357, 1360	\AMCid@checkmax	.. 25, 807–810, 813,
\AMCcompleteMulti .. 12, 63	.. 595, 2244, 2397, 2575	814, 818, 821–824, 827
\AMCcurrentenv 654, 657, 1197	\AMCid@etud	\AMCload@reponse .. 812, 831
\AMCdebutFormulaire 21, 2613	289, 297, 306, 315,	\AMClocalized
\AMCdecimalPoint	327, 330–332, 337,	..... 91
..... 1526, 1707, 1739	352, 473, 2235, 2236,	\AMCloop@k
\AMCdefault@groupmode ..	2261, 2265, 2273,	728, 730, 751, 759, 768
..... 672, 686, 687	2300, 2301, 2303,	\AMCmarginNote
\AMCdontAnnotate . 330, 2132	2308, 2313, 2319,	..... 1064, 1090, 1133
\AMCdontScan	2320, 2340, 2341, 2350	\AMCmem@elt@cat 921, 933, 937
..... 330, 2130	\AMCid@etudfin	\AMCmessage
\AMCdump@reponses . 818, 832	.....	..... 17,
\AMCemptybox 1058, 1083, 1118	... 30, 2302–2304, 2319	279, 591, 991, 1050,
\AMCendAnswer	\AMCid@etudstart	1055, 1184, 1191–
.....	.....	1193, 1212, 1216,
. 1018, 1046, 2115, 2177	..... 29, 2300, 2320	1253–1256, 1294,
\AMCexponent 1528, 1977, 1989	\AMCid@name . 65, 564, 567,	1408, 1611, 1643,
\AMCexternalQuestion .. 1188	569, 573, 575, 915,	1645, 1650, 1652,
\AMCform	972, 1027, 1030, 1041,	1657, 1659, 1668,
..... 16, 863, 2612	1182, 1185, 1212,	1681, 1697, 1789,
\AMCformAfterQuestion ..	1295, 1614, 1617, 2059	1792, 1825, 1827,
..... 846, 1212	\AMCid@quest ... 26, 330–	1847, 1898, 1905,
\AMCformAnswer	332, 564, 567, 569,	1922, 1924, 1927,
..... 845	573, 575, 583, 585,	1929, 1932, 1934,
\AMCformAnswerA ... 862, 966	588, 914, 972, 1027,	1937, 1941, 1944,
\AMCformatChoices	1030, 1041, 1182,	1946, 1953, 1958,
.....	1184, 1185, 1204,	1960, 2000, 2098,
. 1400, 2018, 2164, 2200	1212, 1295, 1615,	2155, 2174, 2191,
\AMCformBeforeQuestion .	1618, 1847, 2098, 2174	2299, 2313, 2415,
..... 845, 851	\AMCIDBoxesA	2419, 2466, 2576–2579
\AMCformBegin . 16, 863, 2613	.... 2236, 2240	\AMCncol@Background
\AMCformFilter	\AMCIDBoxesABC .. 2239, 2262	..... 1536, 1969
..... 984	\AMCIDBoxesB	\AMCncol@Border . 1532, 1969
\AMCformHSpace 66, 861, 2178	.... 2237, 2241	\AMCncol@BorderWidth
\AMCformQuestion	\AMCIDBoxesC	..... 1534, 1968
..... 845	.... 2238, 2241	\AMCneeds@storeboxfalse 2404
\AMCformQuestionA . 849, 973	\AMCids@height . 76, 87, 2261	\AMCneeds@storeboxtrue 2405
\AMCformQuestionN . 848, 856	\AMCids@sidefalse	\AMCnobloc
\AMCformS	... 79, 81	..... 1179, 1283
..... 863	\AMCids@sidetrue	\AMCnoCompleteMulti
\AMCformVSpace	..... 83	.....
..... 66, 845	\AMCids@topfalse	..... 12, 64, 2114
\AMCgroup@pre	.... 79, 83	\AMCNombreCopies
..... 741, 764	\AMCids@toptrue	.....
\AMCgrouploop@next	..... 81	. 2297, 2298, 2592, 2595
.....	\AMCids@width .. 75, 86, 2264	\AMCnoScoreZone . 1163, 1280
766, 779, 797	\AMCidsPosition	\AMCnontextGoto ... 1517, 2009
\AMCgrouploop@prep	..... 73	\AMCnontextSign 1524, 1758, 1759
.....	\AMCidsVar	\AMCnontextVHead .. 1518, 1711
752, 777, 795	..... 77	\AMCnum@copies
\AMCgrouppre@cyclic ... 735	\AMCidsVarN	.....
\AMCgrouppre@fixed	..... 77, 78	... 31, 2297–2299, 2302
..... 697	\AMCif@env .. 655, 1243, 1245	\AMCnum@questions . 819, 821
\AMCgrouppre@withoutreplacement	\AMCifcategory	\AMCnumeric@Hspace
..... 713	..... 947	.....
\AMCgrouppre@withreplacement	\AMCinterBquest	.. 1519, 1531, 1669,
.....	... 71, 1212	1671, 1707, 1726, 1989
704	\AMCinterBrep	
\AMChorizAnswerSep	..... 66, 1031	
.....	\AMCinterIquest	
1035, 1036, 1044	... 70, 1212	
\AMChorizBoxSep	\AMCinterIrep	
.....	..... 66, 1008	
. 1037, 1038, 1041, 1043	\AMCintervalFormat 2029, 2040	
	\AMCintervals	
	17, 2029, 2615	
	\AMClabel	
	..... 2341, 2346	

\AMCnumeric@Vspace . . . . .	\AMCpostOquest . . . . . 72, 2151	\AMCsubsectionNumbered . . . . .
. . . . . 1520, 1530,	\AMCqlabel . . . . . 2345	. . . . . 996, 1003
1642, 1649, 1656,	\AMCquestionaff . . . . . 1164	\AMCsubsectionStar 1000, 1003
1683, 1699, 1713,	\AMCquestionNumberfalse 1281	\AMCsw@p . . . . . 259
1747, 1759, 1976, 1978	\AMCquestionNumbertrue . 45	\AMCsw@p@ . . . . . 261, 263, 265
\AMCnumericChoices . 18, 1400	\AMCrandomseed . . . . . 242	\AMCsz@loggedfalse . . . . . 579
\AMCnumericChoicesPlain . . . . .	\AMCref . . . . . 2341	\AMCsz@loggedtrue . . . . . 592
. . . . . 2016, 2023	\AMCrep@count 833, 835, 837	\AMCtext . . . . . 90
\AMCnumericHide . 2013, 2018	\AMCrep@bloc . . . . . 1007, 1025	\AMCtok@ik . . . . . 718,
\AMCnumericOpts . . . . . 1607	\AMCrep@count . . 564, 567,	719, 721, 726–730,
\AMCnumericShow . 2005, 2018	569, 573, 575, 806,	750, 758, 771, 780, 799
\AMCnumero . . . . . 1165, 2312	815, 835, 837, 841,	\AMCtok@k . . . . .
\AMCocol@Background . . . . .	927, 965, 1006, 1012,	. 663, 678–680, 684, 799
. . . . . 2049, 2056, 2121	1016, 1020, 1023,	\AMCtok@max . . . . . 664
\AMCocol@BoxFrameRule . . . . .	1027, 1030, 1041,	\AMCtok@size . . 665, 753–
. . . . . 2050, 2074, 2121	1049, 1050, 1053–	756, 764, 772, 781, 800
\AMCocol@Foreground . . . . .	1056, 1289, 1293–	\AMCw@termmark . . . 2202, 2246
. . . . . 2052, 2057,	1295, 1401, 1403,	\AMCw@terprint . . 2202, 2246
2126, 2127, 2137, 2140	1610, 1611, 1615,	amcxyfile (environment) 341
\AMCocol@FrameRule . . . . .	1618, 2465, 2466, 2469	\AMCzone . . . . . 346, 347
. . . . . 2051, 2075, 2148	\AMCrep@fini 829, 832, 836, 844	\answer . . . . . 288, 1049, 2433
\AMCopen . . . . . 2047	\AMCrep@init . . 825, 1009,	answers (option) . . . . . 10
\AMCopen@answer . . . . .	1012, 1016, 1020, 1023	asbox (option) . . . . . 10
. 2053, 2060, 2089, 2105	\AMCrep@itemize . 1006, 1025	\association . . . . . 2350
\AMCopen@boxframerule . . . . .	\AMCrep@ligne . . 1012, 1025	auto (environment) . . . . . 2433
. . . . . 2072, 2073, 2120	\AMCrep@o . . . . . 826, 828	automarks (option) . . . . . 11
\AMCopen@boxmargin . . . . .	\AMCrep@perso . . 1016, 1025	
. . . . . 2070, 2071, 2119	\AMCrep@r . . . . . 830	
\AMCopen@contentcommand . . . . .	\AMCrep@tikz . . . . 1020, 1025	
. . . . . 2061, 2149	\AMCrep@tikzmat . . 1023, 1025	
\AMCopen@framerule . . . . .	\AMCreTick . . . . . 330, 2131	
. . . . . 2076, 2077, 2147	\AMCrien@deux . . . . 812, 829	
\AMCopen@Hspace . . 2062,	\AMCscoreZone . . . . 579, 1058	
2063, 2118, 2123, 2135	\AMCscoreZoneAnswerSheet . . . . .	
\AMCopen@LineHeight . . . . .	. . . . . 1058	
. . . . . 2066, 2067, 2088	\AMCsection . . . . . 994	
\AMCopen@Lines . . . . .	\AMCsectionNumbered 994, 1002	
. . . . . 2068, 2069, 2093	\AMCsectionStar . . 998, 1002	
\AMCopen@lines . . . . . 2086	\AMCsetFoot . . . . . 2234	
\AMCopen@lineuptext 2055,	\AMCsetScoreZone . . . . .	
2078, 2103, 2106, 2109	. . . . . 1116, 1117, 1163	
\AMCopen@question . . . . .	\AMCsetScoreZoneAnswerSheet . . . . .	
. 2054, 2059, 2122, 2123	. . . . . 1161–1163	
\AMCopen@Width . . . . .	\AMCshowSignificantDigits . . . . .	
. . . . . 2064, 2065, 2087	. . . . . 1465	
\AMCopenHide . . . . . 2158, 2164	\AMCsignificantDigits . 1460	
\AMCopenOpts . . . . . 2085	\AMCstartWithQuestion . . . . .	
\AMCopenShow . . . . . 2096, 2164	. . . . . 1181, 1189, 1199	
\AMCotextGoto . . . 2047, 2159	\AMCstudentlabel 2341–2344	
\AMCotextReserved 2048,	\AMCstudentNumber . . . . 2235	
2058, 2134, 2138, 2140	\AMCstudentslistfile . . 2353	
\AMCoutsideLabelFormat 519	\AMCsubjectPageTag 2272, 2277	
\AMCpageref . . . . . 2344	\AMCsubsection . . . . . 994	
		<b>B</b>
		\bareme . . . . . 21, 2603
		\baremeDefautM . . . . 21, 2604
		\baremeDefautS . . . . 21, 2605
		\bf . . . . . 1527
		bloc (option) . . . . . 21
		\bonne . . . . . 21, 2601
		\bool . . . . . 939, 1272–1274,
		1296, 1307, 1308,
		1331, 1334, 1338,
		1345, 1347, 1356,
		1600, 1665, 1676,
		1691, 1710, 1715,
		1756, 1760, 1775,
		1788, 1794, 1798,
		1813, 1823, 1824,
		1850, 1854, 1863,
		1904, 1940, 1952, 1970
		box (option) . . . . . 10
		\boxchar . . . . . 337
		\boxput . . . . . 6, 400, 439
		<b>C</b>
		\c . . . . . 944
		calibration (option) . . 5, 10
		catalog (option) . . . . . 10
		\CatalogExterne . . . . . 2549

<code>\champion</code> . . . . .	21, 2614	1623, 1638, 1663,	<code>values-auto</code> . . . . .	2433
<code>\char</code> . . . . .	1420	1674, 1689, 1703,	<code>variable-auto</code> . . . . .	2433
<code>\CheckBox</code> . . . . .	466	1733, 1754, 1783,	<code>variable-multi</code> . . . . .	1178
<code>chiffres</code> (option) . . . . .	21	1845, 2005, 2007,	<code>variable-single</code> . . . . .	1178
<code>\choices</code> . . . . .	2598	2013, 2022, 2025,	<code>\evensidemargin</code> . . . . .	1071, 1073
<code>choices</code> (environment) . . . . .	12, 1004	2026, 2034, 2044,	<code>\examcopy</code> . . . . .	2608
<code>\choicescustom</code> . . . . .	2600	2515, 2516, 2518,	<code>examcopy</code> (environment) . . . . .	2505
<code>choicescustom</code> (environ- ment) . . . . .	12, 1004	2519, 2521, 2522,	<code>\exemplaire</code> . . . . .	21, 2606
<code>\choiceshoriz</code> . . . . .	2599	2525, 2526, 2528, 2529	<code>\exemplairepair</code> . . . . .	2340
<code>choiceshoriz</code> (environ- ment) . . . . .	12, 1004	<code>\csvreader</code> . . . . .	<code>\explain</code> . . . . .	1241, 2495
<code>\choixIntervalles</code> . . . . .	21, 2615	. 2440, 2445, 2451, 2460	<code>\explaincontext</code> . . . . .	1248
<code>\cleargroup</code> . . . . .	14, 787	<b>D</b>	<code>\ExplSyntaxOff</code> . . . . .	
<code>\clist</code> . . . . .	1269, 1275, 1284, 1319, 1332, 1336, 1370, 1379, 1380, 1474, 1477, 1488, 1489, 1704, 1705, 1735, 1736, 1836, 1837	<code>\define@boolkey</code> . . . . .	. . . . .	949, 1399, 1516, 2015, 2028, 2046, 2532
<code>\codeDigitExterne</code> . . . . .	2556, 2557	. . . . .	<code>\ExplSyntaxOn</code> . . . . .	
<code>completemulti</code> (option) . . . . .	10	<code>\define@choicekey</code> . . . . .	. . . . .	863, 1267, 1413, 1521, 2020, 2030, 2513
<code>\coordinate</code> . . . . .	1066	. . . . .	<b>F</b>	
<code>\copieexamen</code> . . . . .	2608	<code>digits</code> (option) . . . . .	<code>\fancypagestyle</code> . . . . .	
<code>copieexamen</code> (environment) . . . . .	21	<code>\ding</code> . . . . .	. . . . .	2248, 2253, 2275, 2282
<code>\copygroup</code> . . . . .	14, 787	<code>\dontannotate</code> . . . . .	<code>\fbox</code> . . . . .	2273
<code>\copygroupfrom</code> . . . . .	14, 787	<code>\donscan</code> . . . . .	<code>\footrulewidth</code> . . . . .	
<code>correc</code> (option) . . . . .	21	<code>\dotfill</code> . . . . .	. . . . .	2251, 2270, 2280, 2285
<code>correcindiv</code> (option) . . . . .	21	<code>\draw</code> . . . . .	<code>\foreach</code> . . . . .	2459
<code>\correctchoice</code> . . . . .		<b>E</b>	<code>\formulaire</code> . . . . .	21, 2612
. . . . .	13, 843, 1049, 2039, 2601	<code>\element</code> . . . . .	<code>\fp</code> . . . . .	1427, 1435, 1437, 1439, 1443, 1448, 1450, 1452, 1456, 1463, 1468, 1475, 1493, 1496, 1834, 1835, 1859, 1860, 1869, 1897, 1898, 1924, 2032, 2033, 2035–2038, 2040, 2041
<code>\CorrigeExterne</code> . . . . .	2533	<code>\endchoices</code> . . . . .	<code>francais</code> (option) . . . . .	10
<code>\CorrigeIndivExterne</code> . . . . .	2533	<code>\endchoicescustom</code> . . . . .	<code>fullgroups</code> (option) . . . . .	11
<code>\crefalias</code> . . . . .	225	<code>\endchoiceshoriz</code> . . . . .	<b>G</b>	
<code>\crefname</code> . . . . .	226	<code>\endcopieexamen</code> . . . . .	<code>\group</code> . . . . .	1419, 1431
<code>\cs</code> . . . . .	876, 880, 881, 883, 886, 889, 892, 896, 901, 903, 906, 908, 911, 913, 917, 918, 920, 923, 924, 926, 929, 930, 932, 935, 944–947, 1277, 1287, 1292, 1317, 1322, 1324, 1327, 1329, 1369, 1373, 1378, 1383, 1388, 1391, 1394–1397, 1415, 1423, 1433, 1436, 1441, 1445, 1457, 1460, 1462, 1465, 1467, 1473, 1487, 1492, 1504, 1595, 1599, 1607, 1609,	<code>\endexamcopy</code> . . . . .	<b>H</b>	
		<code>\endreponses</code> . . . . .	<code>\he@dbas</code> . . . . .	2230, 2255, 2256
		<code>\endreponseshoriz</code> . . . . .	<code>\he@dhaut</code> . . . . .	2231, 2259
		<code>\endreponsesperso</code> . . . . .	<code>\he@dtaille</code> . . . . .	2227, 2230, 2231
		<code>ensemble</code> (option) . . . . .	<code>\hfuzz</code> . . . . .	369
		environments:	<code>\ht</code> . . . . .	367, 1059
		<code>amcxyfile</code> . . . . .	<b>I</b>	
		<code>auto</code> . . . . .	<code>\ifAMC@affichekeys</code> . . . . .	
		<code>choices</code> . . . . .	. . . . .	36, 1205, 1206
		<code>choicescustom</code> . . . . .	<code>\ifAMC@asqbloc</code> . . . . .	42, 846, 852
		<code>choiceshoriz</code> . . . . .	<code>\ifAMC@automarks</code> . . . . .	
		<code>copieexamen</code> . . . . .	. . . . .	60, 978, 2311, 2329, 2336
		<code>examcopy</code> . . . . .		
		<code>question</code> . . . . .		
		<code>question-auto</code> . . . . .		
		<code>questionmult</code> . . . . .		
		<code>Questionnaires</code> . . . . .		
		<code>questionouverte</code> . . . . .		
		<code>reponses</code> . . . . .		
		<code>reponseshoriz</code> . . . . .		
		<code>reponsesperso</code> . . . . .		
		<code>tikz-multi</code> . . . . .		
		<code>tikz-single</code> . .		

<code>\ifAMC@calibration</code> . . . . .	<code>\ifAMC@printkeyoutside@</code>	1444, 1472, 1475,
46, 294,	538, 546	1477–1479, 1481,
303, 312, 325, 330–	<code>\ifAMC@qbloc</code> . . . 41, 1173,	1494, 1497, 1499,
332, 335, 1050, 1055,	1204, 1212, 1238,	1505, 1506, 1508,
1183, 1190, 1215,	1240, 2010, 2160, 2196	1509, 1511, 1601,
1253–1256, 1294,	<code>\ifAMC@rbloc</code> . 43, 1007, 1010	1621, 1624, 1626–
1318, 2307, 2313,	<code>\ifAMC@shuffleG</code> 33, 745, 748	1628, 1630, 1639,
2349, 2354, 2466, 2580	<code>\ifAMC@survey</code> . . . 2408, 2433	1664, 1675, 1682,
<code>\ifAMC@catalog</code> . . . . . 47, 386	<code>\ifAMC@textPos</code> . 39, 1027,	1688, 1690, 1692,
<code>\ifAMC@correc</code> . . . . .	1030, 1040, 1204, 1212	1698, 1704, 1706,
38, 378, 1052, 1249,	<code>\ifAMC@watermark</code> . . . . .	1711, 1717, 1721,
1613, 2089, 2105, 2468	51, 2246, 2566	1725, 1728, 1735,
<code>\ifAMC@correchead</code> . . . . .	<code>\ifAMC@zoneformulaire</code> . .	1737, 1743, 1746,
. . . . 35, 1242, 2246,	. . . . 57, 526, 563, 867	1749, 1761, 1786,
2289, 2304, 2561, 2567	<code>\ifAMC@complete@multi</code> 44, 840	1787, 1790, 1793,
<code>\ifAMC@draw@cross</code> . . . . .	<code>\ifAMC@formulaire@dedans</code>	1797, 1800, 1806,
. . . . . 362, 401, 422	56, 525,	1810, 1811, 1816,
<code>\ifAMC@ensemble</code> . . . . .	566, 582, 1846, 1999,	1817, 1838–1840,
. . . . . 54, 524, 541,	2097, 2154, 2173, 2190	1843, 1851, 1854,
562, 581, 866, 978,	<code>\ifAMC@ids@side</code> . . . . 74, 2264	1855, 1861, 1866,
981, 985, 990, 1206,	<code>\ifAMC@ids@top</code> . . . . 73, 2261	1868, 1872, 1885,
1406, 1846, 1998,	<code>\ifAMC@needs@storebox</code> . . .	1886, 1891, 1899,
1999, 2097, 2102,	. . . . 2404, 2411, 2422	1900, 1906, 1908,
2116, 2122, 2154,	<code>\ifAMC@questionNumber</code> 45, 1171	1911, 1919, 1926,
2173, 2178, 2181,	<code>\ifAMC@sz@logged</code> . . . 579, 590	1927, 1931, 1932,
2190, 2311, 2576, 2586	<code>\ifAMC@type@multi</code> . . . 50, 840	1934, 1947, 1954,
<code>\ifAMC@extractOnly</code> . . . . .	<code>\ifAMC@une@bonne</code> . . . . 49, 842	1955, 1961, 1975, 1988
. . . . . 40, 1966, 2590	<code>\ifcase</code> . . . . . 78	
<code>\ifAMC@fullGroups</code> . . 34, 754	<code>\ifcsname</code> . . . 1102, 1107,	<b>K</b>
<code>\ifAMC@inside@box</code> . . . . .	1112, 1147, 1152, 1157	<code>\keys</code> . . . . . 1303, 1330,
. . . . . 52, 544, 572,	<code>\ifcsvstrcmp</code> . . . . .	1523, 1596, 1849, 2008
2117, 2179, 2577, 2587	. 2446, 2452, 2453, 2461	<code>keys (option)</code> . . . . . 10
<code>\ifAMC@inside@digit</code> . 55, 478	<code>\ifdim</code> . . . . . 1076	<b>L</b>
<code>\ifAMC@invisible</code> . . . 61, 292	<code>\ifKV@AMC@BoxOnly@ordered</code>	<code>\lastchoices</code> . . . . . 825, 2611
<code>\ifAMC@keepmemory</code> 988, 2317	2182	<code>\lastxpos</code> . . . . . 2519
<code>\ifAMC@keyslines</code> 37, 1205, 1206	<code>\ifKV@AMC@dim@cross</code> . . . . .	<code>\lastypos</code> . . . . . 2522
<code>\ifAMC@ordre</code> . . . . . 32, 826	382, 397, 414	<code>\linebreak</code> . . . . . 2112
<code>\ifAMC@outside@box</code> . . . . .	<code>\ifKV@AMC@Open@annotate</code> 2132	
53, 524, 542, 2578, 2588	<code>\ifKV@AMC@Open@dots</code> . . . . 2090	<b>M</b>
<code>\ifAMC@pagelayout</code> . . . . .	<code>\ifKV@AMC@Open@lineup</code> . . .	<code>\m@rque</code> . . . . 2226, 2255–2258
. . . 58, 2243, 2288, 2560	2101, 2144	<code>\m@rqueCalage</code> . . . 2225, 2226
<code>\ifAMC@pdfform</code> . . . . .	<code>\ifKV@AMC@Open@retick</code> . . 2131	<code>\marginpar</code> . . . . . 1087, 1129
62, 350, 2305, 2321, 2414	<code>\ifKV@AMC@Open@scan</code> . . . 2130	<code>\mauvaise</code> . . . . . 21, 2602
<code>\ifAMC@plain</code> . . . . . 48, 2497	<code>indivanswers (option)</code> . . 6, 10	<code>\melangegroupe</code> . . . . 21, 2609
<code>\ifAMC@postcorrect</code> . . . . .	<code>init (option)</code> . . . . . 10	<code>\MessageBreak</code> . . . . . 2495
. . . . 59, 843, 2579, 2589	<code>\InputIfFileExists</code> . . . . 15	<code>\multiSymbole</code> 1173, 1214, 1232
<code>\ifAMC@printformoutside</code>	<code>\insertgroup</code> . . 14, 744, 2610	
. . . . . 522, 531	<code>\insertgroupfrom</code> . . . 14, 744	<b>N</b>
<code>\ifAMC@printformoutside@</code>	<code>insidebox (option)</code> . . . . . 10	<code>\namefield</code> . . . 15, 347, 2614
. . . . . 521, 528	<code>\int</code> 878, 880, 884, 890, 897,	<code>\namefielddots</code> . . . . . 348
<code>\ifAMC@printkeyoutside</code> .	936, 1271, 1284, 1323,	<code>\newbox</code> . . . . . 363
. . . . . 539, 549	1325, 1334, 1340,	<code>\newline</code> . . . . . 1205
	1341, 1364, 1428,	<code>\newsavebox</code> . . . . . 359

\newstorebox ..... 2423  
 \nobreak ..... 520, 2145  
 \node ..... 1032, 1033, 1228  
 nopage (option) ..... 6, 11  
 noshuffle (option) ..... 10  
 noshufflegroups (option) . 10  
 \nouveaugroupe .... 663, 788  
 nowatermark (option) .. 4, 10  
 \NoWatermarkExterne ... 2533

## O

\oddsidemargin .. 1070, 1074  
 \onecopy 2296, 2435, 2509, 2606  
 options:

answers ..... 10  
 asbox ..... 10  
 automarks ..... 11  
 bloc ..... 21  
 box ..... 10  
 calibration ..... 5, 10  
 catalog ..... 10  
 chiffres ..... 21  
 completemulti ..... 10  
 correc ..... 21  
 correcindiv ..... 21  
 digits ..... 10  
 ensemble ..... 21  
 francais ..... 10  
 fullgroups ..... 11  
 indivanswers ..... 6, 10  
 init ..... 10  
 insidebox ..... 10  
 keys ..... 10  
 nopage ..... 6, 11  
 noshuffle ..... 10  
 noshufflegroups .... 10  
 nowatermark ..... 4, 10  
 ordre ..... 21  
 outsidebox ..... 10  
 pdfform ..... 11  
 plain ..... 11  
 postcorrect ..... 11  
 separateanswersheet .  
 ..... 5, 10, 16  
 storebox ..... 11  
 \or ..... 80, 82  
 ordre (option) ..... 21  
 outsidebox (option) ..... 10  
 \ouverte@vs ..... 1178

## P

\PackageError .....  
 .... 648, 690, 761, 2494  
 \pageref ..... 2344

pdfform (option) ..... 11  
 \pdfpageheight 329, 2528, 2529  
 \pdfpagewidth 329, 2525, 2526  
 \pgfextractx ..... 1067  
 \pgfpointanchor ..... 1068  
 \pgfpointdiff ..... 1067  
 \pgfpointorigin ..... 1067  
 plain (option) ..... 11  
 postcorrect (option) .... 11  
 \prg ..... 865, 868,  
 870, 873, 1380, 1476, 1489

## Q

\question ..... 1178, 1185  
 question (environment) ..  
 ..... 11, 1178  
 question-auto (environ-  
 ment) ..... 2433  
 \QuestionIndicative .....  
 ..... 13, 1253, 1288  
 questionmult (environ-  
 ment) ..... 11, 1178  
 Questionnaires (environ-  
 ment) ..... 2433  
 questionouverte (environ-  
 ment) ..... 1178

## R

\raisebox ..... 1527  
 \refstepcounter ..... 1171  
 \reponses ..... 2598  
 reponses (environment) .. 21  
 \reponseshoriz ..... 2599  
 reponseshoriz (environ-  
 ment) ..... 21  
 \reponsesperso ..... 2600  
 reponsesperso (environ-  
 ment) ..... 21  
 \restituegroupe ... 21, 2610  
 \retick ..... 332  
 \rngstate ..... 2308

## S

\savebox ..... 360  
 \savepos ..... 2516  
 \sbox ..... 1059  
 \scoring ..... 13,  
 1220, 1226, 1253, 2603  
 \scoringDefaultM .....  
 ..... 13, 1253, 2604  
 \scoringDefaultS .....  
 ..... 13, 1253, 2605  
 \ScoringExterne ..... 2533  
 \section ..... 995, 999

separateanswersheet (op-  
 tion) ..... 5, 10, 16  
 \seq ..... 1270, 1344,  
 1352, 1418, 1426–1428  
 \setdefaultgroupmode .. 686  
 \setgroupmode ..... 672, 686  
 \shuffle@it ..... 818, 832  
 \shufflegroup .....  
 . 14, 710, 722, 744, 2609  
 \shufflegroupslice .....  
 ..... 727, 730, 747  
 \smash ..... 2089, 2105  
 \space ..... 2495  
 \storebox ..... 2424  
 storebox (option) ..... 11  
 \str ..... 933  
 \strut ..... 1077, 1079  
 \SujetExterne ..... 2533

## T

\tex ..... 1421  
 \textasciicircum ..... 1529  
 \TextField ..... 352  
 \textit ..... 99,  
 112, 139, 153, 192, 218  
 \textsc ..... 2249  
 \textsf ..... 504  
 \thecsvrow .....  
 . 1032, 1033, 2474, 2483  
 tikz-multi (environment) 1004  
 tikz-single (environment)  
 ..... 1004  
 \tikzset ..... 2472  
 \times ..... 1529  
 \tl 898, 899, 904, 909, 938,  
 940, 1415, 1417, 1424,  
 1425, 1622, 1625,  
 1631, 1634, 1785,  
 1799, 1801, 1804,  
 1805, 1809, 1814,  
 1815, 1817, 1820,  
 1821, 1829, 1841, 1842  
 \tracepos .... 296, 305, 314

## U

\une@rep .... 1006, 1007,  
 1012, 1016, 1020,  
 1023, 1052, 1056, 2468  
 \useasboundingbox .... 417  
 \usebox ..... 361  
 \usestorebox ..... 2425  
 \usetikzlibrary ..... 2471

V		
<code>values</code> (environment) . . .	<u>2433</u>	
<code>values-auto</code> (environment)		
	<u>2433</u>	
<code>variable-auto</code> (environ-		
ment) . . . . .	<u>2433</u>	
<code>variable-multi</code> (environ-		
ment) . . . . .	<u>1178</u>	
		<code>variable-single</code> (environ-
		ment) . . . . .
		<u>1178</u>
		<code>\version</code> . . . . .
		2583
		<code>\vfuzz</code> . . . . .
		369
		<code>\vrule</code> . . . . .
		1771
		<code>\vtop</code> . . . . .
		1347
		W
		<code>\wd</code> . . . . .
		369, 370, 1059
		<code>\with</code> . . . . .
		1319, 2355, 2356,
		2584–2591, 2593, 2595
		<code>\wrongchoice</code> . . . . .
		13,
		842, 843, <u>1049</u> , 2039, 2602
		X
		<code>\x</code> . . . . .
		2459, 2461
		<code>\xyclose</code> . . . . .
		344
		<code>\xyopen</code> . . . . .
		342