

Auto Multiple Choice

Design of MCQ tests with automated correction and grading <http://home.gna.org/auto-qcm/>

Copyright © 2008-2013 Alexis Bienvenüe

This document can be used according to the terms of the GNU General Public License, version 2 or later.

COLLABORATORS

	<i>TITLE :</i> Auto Multiple Choice		
<i>ACTION</i>	<i>NAME</i>	<i>DATE</i>	<i>SIGNATURE</i>
WRITTEN BY	Alexis Bienvenüe	June 29, 2016	
Translation from French	Jean Bérard	June 29, 2016	
Translation from French	Georges Khaznadar	June 29, 2016	

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

1	License	1
2	Prerequisites	1
2.1	Operating system	1
2.2	Software	1
2.2.1	Recommended versions for software used by AMC	1
2.3	LaTeX	1
3	Usage notes	2
3.1	Numerical limitations	2
3.2	Versions	2
3.3	Installation	2
3.4	Access	2
4	AMC-TXT syntax	3
4.1	Comments	3
4.2	General options	3
4.2.1	Separate answer sheet	5
4.3	Questions	5
4.3.1	Questions options	5
4.3.2	Scoring strategy	5
4.3.3	Open questions	6
4.4	Multi-line	6
4.5	Bold, italic	7
4.6	Images	7
4.7	Pieces of LaTeX code	7
4.8	Arabic language	7
4.9	Japanese language	8
5	LaTeX source file	8
5.1	A simple example	8
5.2	An example with groups of questions and shuffling	10
5.3	An example with a separate answer sheet	11
5.4	Description of the LaTeX commands	14
5.4.1	Package options	14
5.4.2	Description of a copy	15
5.4.3	Questions and answers	15
5.4.4	Putting answers on multiple columns	16

5.4.5	Groups of questions	17
5.4.6	Page size, margins	18
5.4.7	Check-box presentation style	18
5.4.8	Questions presentation style	19
5.4.9	Separate answer sheet presentation style	20
5.4.10	Code acquisition	20
5.4.11	Open questions	21
5.4.12	Choice of shuffling parameters	22
5.4.13	Using sectioning and separate answer sheet	22
5.4.14	Using references inside the test sheets	22
5.4.15	Customizing some texts inserted by AMC	23
5.5	Options for AMC	23
5.6	Mathematical questions with randomized statements	23
5.6.1	Using package fp	23
5.6.1.1	Choosing an interval	24
5.6.1.2	Coding the result	24
5.6.2	Using package pgf/tikz	26
5.6.2.1	A simple computation	26
5.6.2.2	Graphics	27
5.6.3	Using LuaLaTeX	27
6	Usage - Graphical interface	28
6.1	Creating a new project and subject	28
6.2	Preparing the subject	29
6.3	Printing and exam	29
6.4	Reading the copies	30
6.4.1	Automated input	30
6.4.2	Manual input	30
6.5	Correction	30
6.5.1	Process	31
6.5.2	Scoring strategy	31
6.5.3	Global scoring strategy	34
6.5.4	From scoring strategy to students marks	34
6.6	Identification of the students	35
6.6.1	List of the students	35
6.6.2	Association	35
6.6.2.1	Automated association	35
6.6.2.2	Manual association	36
6.7	Exporting the scores list	36
6.7.1	Export to ODS (OpenOffice, LibreOffice)	36
6.8	Annotation	36

7	Alternative usages	37
7.1	Photocopied subject	37
7.2	Post correcting	38
7.3	Nominative sheets	39
8	Commands manual	40
8.1	auto-multiple-choice	40
8.2	AMC-prepare	41
8.3	AMC-meptex	42
8.4	AMC-imprime	42
8.5	AMC-getimages	43
8.6	AMC-analyse	43
8.7	AMC-note	44
8.8	AMC-association-auto	45
8.9	AMC-association	46
8.10	AMC-export	46
8.11	AMC-annotate	47
8.12	AMC-regroupe	49
8.13	AMC-mailing	50

Abstract

AMC is a set of utilities allowing the use of MCQ written in plain text or LaTeX, and their automated correction and grading from scans of the answer sheets using Optical Mark Recognition (OMR). This document describes its use (for the version 1.2.1~svn 1403).

The AMC project is hosted by [Gna !](http://home.gna.org/auto-qcm/) : you can find the AMC web-site there <http://home.gna.org/auto-qcm/>, and the web-page where the project is hosted there <https://gna.org/projects/auto-qcm/>.

License

Auto Multiple Choice is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 2 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details. You should have received a copy of the GNU General Public License along with this program. If not, see <http://www.gnu.org/licenses/>.

Prerequisites

Operating system

The AMC utilities have been written for Linux. They might be adapted to MAC OS X or Windows without too many difficulties, but I won't do that in the near future. Please contact me if you are interested to do it : I can help you a little, and I will include changes in AMC if necessary.

Software

Note

If you install AMC with deb or RPM package (on debian, Ubuntu and derivatives, Mandriva, openSUSE, Fedora), every software used by AMC will be automatically installed.

The main packages/software that are required for the use of AMC are the following:

- LaTeX
- ImageMagick
- Perl, together with Gtk2-Perl and Glade::XML for the graphical user interface

Recommended versions for software used by AMC

A few bugs in old versions of software used by AMC are particularly detrimental to its operating normally:

- With old versions of Net : : CUPS (debian package `libnet-cups-perl`), the command for printing the test sheets leads to a premature exit from the AMC graphical user interface. This bug is fixed in versions 0.61 and later of Net : : CUPS.
- With old versions of ImageMagick, the merging of corrected answer sheets into a single PDF file per student does not work when answer sheets comprise more than one page. This bug is fixed in versions 6.5.5 and later of ImageMagick.

LaTeX

From version 1.1 of AMC, it is not mandatory anymore to write your questionnaires using LaTeX language (see Section 4 for a plain text alternative syntax). However, LaTeX is still the native AMC language for questionnaires descriptions, and allows the user to design his questionnaires with tremendous freedom. The templates that are provided should allow those who are not deterred by LaTeX format to quickly start writing their own MCQs.

Usage notes

Numerical limitations

In the present version (starting with version 0.156), the maximum number of students per test is limited to 4095, and the maximum number of pages per test is limited to 63.

The maximum number of answers for a given question is limited to 199.

LaTeX's own limitations may make compilation impossible, producing e.g. a « No room for a new \toks » error. In that case, try again using the package `etex`, thanks to the `\usepackage{etex}` command.

Versions

Even though it should not have too many consequences, it is recommended not to change the program version while working with a given project (between the time when the test sheets are printed and the time when tests are corrected and graded). There are a few modifications which may interfere with AMC operating normally if you apply them while working with a given project:

- If you have compiled and printed with a version of AMC earlier than 0.155, and then handle the project with version 0.156 or later, add the option `versionA` in the reference to the package `automultiplechoice` in the LaTeX file. Example:

```
\usepackage[box,completemulti,versionA]{automultiplechoice}
```

- From version 0.262 on, the drawing of the checkboxes changed to be slightly larger, which may alter the layout of the test sheets.
- From version 0.267, if you want to produce A4 format test sheets, you have to state it explicitly in the LaTeX file:

```
\documentclass[a4paper]{article}
```

- From version 0.394, if your LaTeX code needs package `graphicx`, you have to load it explicitly (earlier versions of AMC loaded it).
- The LaTeX command **\AMCcode** was rewritten in version 0.518. It is now more robust and configurable, but you will have to adapt the LaTeX sources you wrote for older versions to keep the same layout.
- From version 1.1.0, all AMC data is stored in SQLite databases instead of lots of XML files. The first time you open a project, old XML files will be converted to the new format.

Installation

With debian, Ubuntu, or a derived operating system, you can use the debian official repository and ubuntu AMC repositories. On Mandriva, openSUSE or Fedora, you can use prebuilt RPM packages. The installation procedure is described on the [AMC web-site](#).

You can also download the source code archive on the project [download space](#), then use the following commands in a terminal:

```
tar xvzf auto-multiple-choice_xxxx_precomp.tar.gz
cd auto-multiple-choice-xxx
make
sudo make install
```

Access

Once the installation is made, the graphical user interface can be launched by choosing Applications → Education → Auto Multiple Choice in the Gnome general menu (or its KDE or other analog), but one can also use the **auto-multiple-choice** command.

AMC-TXT syntax

For users who are not ready to learn LaTeX (if not already speaking this language), AMC includes a filter to process simple plain text files in a particular format, named AMC-TXT. This section is devoted to detail the syntax of AMC-TXT files. If you prefer use LaTeX to gain a fine-control over your questionnaires, skip to next section.

Let us begin with a simple exemple:

```
# AMC-TXT source file
Title: My first AMC questionnaire

Presentation: Please answer the following questions
the best you can.

* What is the capital city of Cameroon?
+ Yaounde
- Douala
- Kribi

** From the following numbers, which are positive?
- -2
+ 2
+ 10
```



Important

The file that contains your questionnaire must be a plain text file, UTF-8 encoded. This is the default encoding of several text editors, like **gedit**. Don't use a text editor that can format your text with bold, images, and so on, like OpenOffice/LibreOffice, or equivalent: these save your text with a lot of other data, and AMC won't be able to read it.

Note

The default font that will be used is **libertine**, an open source font that contains chracaters for a lot of languages. You have to install this font or choose another one (see options below). If you install AMC with a debian/ubuntu package, this font will be install together as a recommended package.

You already understand AMC-TXT structure: some general options first, and then questions.

Comments

You can write some comments inside your AMC-TXT source file in lines *beginning* with a `#'. These lines won't be considered by AMC.

General options

Here are the options you can use (in any order):

Lang: use it to specify a language the questionnaire is written in. At present, only DE (german), ES (spanish), FR (french), IT (Italian), NL (Dutch), NO (Norwegian), JA (Japanese, see Section 4.9) and AR (arabic, see Section 4.8) are supported. Without this option, english will be selected. You can also define the localized string used for another language (see the L-xxx options).

PaperSize: Sets the paper size. Possible values are A3, A4, A5, A6, B3, B4, B5, B6, letter, legal, ANSIA, ANSIB, ANSIC, ANSID, ANSIE.

Title: The exam title, written on top of the sheet.

Presentation: A text that presents the exam (length, rules...).

ShuffleQuestions: If 1 (default), questions will be shuffled so that their order is different from one sheet to the other. If 0, the questions will allways show with the same order as in your file.

Code: Give a positive integer value n to add boxes so that students will be able to code their student number on their sheets, with n digits.

Columns: Give a positive integer value n to get a subject with n columns.

CompleteMulti: If 1 (default), for multiple questions (those for which zero, one or several answers are correct), an answer "None of these answers are correct" will be added. Without it, it should be impossible to make a difference between "the student didn't answer for this question" and "the student thinks no answer is correct for this question". If you don't want this answer to be added, set this option to 0.

L-None: Give a string to replace *None of these answers are correct* (see previous option).

QuestionBlocks: If 1 (default), all questions will be enclosed in a invisible frame that prevents it to be splitted accross several columns or pages. If 0, questions are allowed to be splitted if necessary: that can save pages at the cost of readability.

L-Question: Give a translation of *Question* in your questionnaire, if you need.

L-Name: Give a translation of *Name and surname*, a text written in the box where students are to write their udentity.

L-Student: Small text that asks students to code their student numbers and write their name, when `Code` option is used.

ManualDuplex: If 1 (this is not the default value), each subject will consist of an even number of pages, so that the user can manually print the PDF subject for all students in one go in duplex mode.

SingleSided: If 1 (this is not the default value), no blank page will be added between the question and the separate answer sheet, even if the question has an odd number of pages. This mode can be useful when the subjects are printed single-sided, or when it is not necessary to separate question and answer sheet.

BoxColor: Color of the boxes to be filled by the students. This 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). The color has to be given as a valid **xcolor** color (see **xcolor** LaTeX package documentation for details), such as *red*, *magenta*, *pink*, *lightgray*, *cyan*, or in the form `#RRGGBB`, like `#FFBEC8` for some light red.

DefaultScoringS: Default scoring strategy for simple questions (questions for which one and only one anser is correct). See Section 6.5.2 for details. The default value gives one point for the right answer, and zero for others.

DefaultScoringM: Default scoring strategy for multiple questions (questions for which there can be zero, one or several correct answers). See Section 6.5.2 for details. The default value is `haut=2`, so that a perfect answer gives 2 points, and each error (ticking a box that should not be ticked, or leaving a box that should be ticked unticked) takes one point off (keeping the score non-negative).

LaTeX: Set this option to 1 if you want to use LaTeX commands in your texts. This allows for exemple to insert mathematical formulas, like `\sqrt{a+b}`. If 0 (default), all your texts will be written unmodified.

LaTeX-Preamble: Give commands you want to be added to the LaTeX preamble (for exemple `\usepackage` commands).

LaTeX-BeginDocument: Give commands to be inserted at the beginning of the LaTeX **document** environment (for exemple macro definitions).

Disable: Gives a comma-separated list of features to disable. Current implemented features are `images` (see Section 4.6), `embf` (see Section 4.5) and `local_latex` (see Section 4.7).

Separate answer sheet

To use separate answer sheets for your questionnaire, consider the following options:

SeparateAnswerSheet: If 1, a separate answer sheet will be added.

AnswerSheetTitle: Title of the answer sheet.

AnswerSheetPresentation: Presentation of the answer sheet. For example, remind the students that the answers are to be given on this sheet *only*.

AnswerSheetColumns: Number of columns for the answer sheet.

Questions

Simple questions (questions for which one and only one answer is correct) begin with a `*` at the beginning of the line, and multiple questions (questions for which there can be zero, one or several correct answers) begin with a `**`. Insert then the question itself, and the choices, introduced with a `+` for correct ones and with a `-` for wrong ones.

Questions options

Some options are available for questions. They must be separated by commas and enclosed by square brackets just after the leading `*`'s, as in the following example:

```
*[ordered,horiz,name=sum] How much are
one plus one?
- 0
- 1
+ 2
```

Available options for questions:

horiz present choices horizontally, not on separated lines.

columns=*n* make *n* columns for the choices.

ordered don't shuffle the choices, keep the same order as in the description file.

id=*xxxx* Give a name to the question, so as to locate the corresponding results easily in the exported spreadsheets. This name must contain only simple characters, without accents and LaTeX special characters such as `_`, `^`, `%`...

`name` can also be used instead of `id` for compatibility with old versions, but you should prefer using `id`.

indicative don't use this question score to compute the global student's score.

Scoring strategy

You can set the scoring strategy for a particular question or choice enclosing it with braces just after the leading characters (`*`, `**`, `+` or `-`) and the possible options, as in the following example. See Section 6.5.2 for details about the scoring strategy syntax.

```
{b=2,m=-1} What is the capital city of France?
+ Paris
- Lille
- Marseille
- Ouagadougou
-{-2} New York

**[ordered,horiz,name=positive]{haut=1} From the following numbers, which are positive?
- -2
+ 2
+ 10
```

Open questions

You can define open question giving options (see Section 5.4.11) enclosed with < and >, as in the following example:

```
*<lines=4> Describe the moon.
-[O]{0} O
-[P]{1} P
+[V]{2} V
```

You should also consider using the following global options:

L-OpenText: Text used to tell the student to write the answer on the separate answer sheet (if relevant).

L-OpenReserved: Text to be written along the open questions boxes to tell the student not to consider these boxes.

Multi-line

You can always continue some text on the next lines (even if some of them are empty), provided that these following lines cannot be considered as the beginning of an option definition, of a question or of a choice. As an example, consider the following question:

```
* How much are 2
+ 2?
- 0
+ 4
- 10
```

This is a correct AMC-TXT question. However, it won't be treated as you'd like to, because here the second line is not considered as being the following of the first one, but form the first choice of the question!

A similar problem arises from the following AMC-TXT question, where `Gershwin:` is considered as a general option definition...

```
* You all know Georges
  Gershwin: he is a composer.
  When was he born?
+ in 1898
- in 1892
- in 1902
```

This is a correct way to write it:

```
* You all know Georges Gershwin:
  he is a composer.
  When was he born?
+ in 1898
- in 1892
- in 1902
```

Note that line breaks can be inserted leaving an empty line, as:

```
Presentation: Title

Description of the exam.

** Difficult question.

How many stars in the sky?
- one
- two
- ten millions
```

Bold, italic

To get *bold* text, enclose it between [`*` and `*`]. To get *italic* text, enclose it between [`_` and `_`].

```
* What is the [*capital city*] of [_France_]?  
+ Paris  
- Lille  
- Marseille
```

Images

You can add images in your document using the following syntax:

```
![height=2cm]images/bird.png!
```

Here, the image `images/bird.png` that is located in the project directory will be appened with 2cm height. Options that can be used inside the square brackets are those from the `\includegraphics` LaTeX command (*`width=3cm` or `keepaspectratio`* for example). To get a centered image that is three quarters of the line width, use

```
!{center}[width=.75\linewidth]images/map.pdf!
```

Pieces of LaTeX code

You can include some small parts of LaTeX code in your document, including it in double square brackets, as in:

```
Questions with a [[\multiSymbole{}]] may have zero, one or more right answers.
```

Arabic language

Writting a questionnaire in arabic is a little special. Use of course option

```
Lang: AR
```

You can also consider the following global options:

ArabicFont: This is the font used for arabic text. Default value is *Rasheeq*, a font from the project [ArabEyes](#) (you can find it on [debian/ubuntu](#) in the `ttf-arabeyes` package).



Warning

For better working and portability, give an explicit value for `L-Name` and `L-Student` options.

Note

To insert texts with non-arabic characters, you must turn on `LaTeX` option and enclose these texts as a `\textLR` LaTeX command argument, as in `\textLR{xelatex command}`.

Japanese language

Japanese language questionnaires can be produced with option

Lang: JA

AMC will make some adjustments on the produced LaTeX source to allow Japanese characters to be included.



Warning

In this case, AMC will use **pTeX** to process the LaTeX file made from your AMC-TXT source file. AMC needs a *recent* version of **pTeX** to work. Versions of pTeX from texlive 2009, that can be found on old versions of some linux distributions, are *not* compatible.

LaTeX source file

This section describes the LaTeX commands that allows you to design your exam answer sheets from a LaTeX source file. If you chose another format for your source file, you can skip this section. The MCQ can be described as a LaTeX file using the `automultiplechoice` package. You can check the LaTeX file you are designing at any moment by compiling it with the **latex** command, then visualizing the resulting dvi file.

We start with a few examples giving quick illustrations of how to build LaTeX files for MCQs; the corresponding tex file are available as templates, so that one can create a new MCQ project starting with one of these templates.

A simple example

```

1 \documentclass[a4paper]{article}
2
3 \usepackage[utf8x]{inputenc}      ❶[1]
4 \usepackage[T1]{fontenc}
5
6 \usepackage[box,completemulti]{automultiplechoice}  ❷[2]
7 \begin{document}
8
9 \onecopy{10}{      ❸[3]
10
11 %%% beginning of the test sheet header:      ❹[4]
12
13 \noindent{\bf QCM \hfill TEST}
14
15 \vspace*{.5cm}
16 \begin{minipage}{.4\linewidth}
17 \centering\large\bf Test\ Examination on Jan., 1st, 2008\end{minipage}
18 \namefield{\fbox{      ❺[5]
19 \begin{minipage}{.5\linewidth}
20 \quad Firstname and lastname:
21
22 \quad \vspace*{.5cm}\dotfill
23 \quad \vspace*{1mm}
24 \end{minipage}
25 }}
26
27 \begin{center}\em
28 Duration : 10 minutes.
29
30 No documents allowed. The use of electronic calculators is forbidden.
31
```

```

32
33 Questions using the sign \multiSymbole{} may have
34 zero, one or several correct answers. Other questions have a single correct answer.
35
36 Negative points may be attributed to \emph{very
37 bad} answers.
38 \end{center}
39 \vspace{1ex}
40
41 %%% end of the header
42
43 \begin{question}{prez} ❸[6]
44 Among the following persons, which one has ever been a President of the French Republic?
45 \begin{choices}
46 \correctchoice{René Coty}
47 \wrongchoice{Alain Prost}
48 \wrongchoice{Marcel Proust}
49 \wrongchoice{Claude Monet}
50 \end{choices}
51 \end{question}
52
53 \begin{questionmult}{pref} ❷[7]
54 Among the following cities, which ones are french prefectures?
55 \begin{choices}
56 \correctchoice{Poitiers}
57 \wrongchoice{Sainte-Menehould}
58 \correctchoice{Avignon}
59 \end{choices}
60 \end{questionmult}
61
62 \clearpage ❹[8]
63
64 } ❹[9]
65
66 \end{document}

```

A few extra details on this example:

- ❶ The packages `inputenc` and `fontenc` allow one to use the UTF-8 encoding. You can of course modify them depending on the encoding you want to use.
- ❷ The options used here for the `automultiplechoice` LaTeX package prevent questions from being split between two pages (`box`) and to automatically complete any multiple choice question by a standard answer allowing the student to mention that, in her/his opinion, none of the listed answers is correct (`completemulti`).
- ❸ The **onecopy** command produces as many (distinct) realizations of the MCQ test as desired (here 10). See Section 5.4.2 for an alternative syntax using an environment.
- ❹ Lines that start here describe the header of the test-sheet.
- ❺ The **namefield** command specifies where students write their name.
- ❻ The environments **question** and **choices** allow one to build a multiple choice question for which there is a single correct answer. A unique identifier for the question has to be specified (here: *prez*).
- ❼ The environments **questionmult** and **choices** allow one to build a multiple choice question that may have zero, one or several correct answers. Student are asked to check all the boxes corresponding to an answer that she/he thinks is correct, or the last box (added automatically thanks to the `completemulti` option used in the reference to the package in line 6).
- ❽ The **clearpage** is used to start the next realization of the MCQ test on a new page.
- ❾ This marks the end of the **onecopy** command (started at line 9).

An example with groups of questions and shuffling

In this example, we want the order in which questions appear in the MCQ test to be different from one realization of the test to the other, but still keeping together questions dealing with the same subject. To this end, we create two groups of questions, within which questions are shuffled at random.

```

1 \documentclass[a4paper]{article}
2
3 \usepackage[utf8x]{inputenc}
4 \usepackage[T1]{fontenc}
5
6 \usepackage[box,completemulti]{automultiplechoice}
7 \begin{document}
8
9 %%% preparation of the groups
10
11 \element{geographie}{
12   \begin{question}{Paris}
13     In which continent is Paris?
14     \begin{choices}
15       \correctchoice{Europe}
16       \wrongchoice{Africa}
17       \wrongchoice{Asia}
18       \wrongchoice{planet Mars}
19     \end{choices}
20   \end{question}
21 }
22
23 \element{geographie}{
24   \begin{question}{Cameroon}
25     Which is the capital city of Cameroon?
26     \begin{choices}
27       \correctchoice{Yaoundé}
28       \wrongchoice{Douala}
29       \wrongchoice{Abou-Dabi}
30     \end{choices}
31   \end{question}
32 }
33
34 \element{histoire}{
35   \begin{question}{Marignan}
36     In which year did the battle of Marignan take place?
37     \begin{choiceshoriz}
38       \correctchoice{1515}
39       \wrongchoice{1915}
40       \wrongchoice{1519}
41     \end{choiceshoriz}
42   \end{question}
43 }
44
45 \element{histoire}{
46   \begin{questionmult}{Nantes}
47     What can be said about the \emph{Édit de Nantes}?
48     \begin{choices}
49       \correctchoice{It was signed in 1598}
50       \correctchoice{Il has been revoked by Louis XIV}
51       \wrongchoice{It was signed by Henri II}
52     \end{choices}
53   \end{questionmult}
54 }
55
56 %%% copies

```

```

57
58 \onecopy{10}{
59
60 %%% beginning of the test sheet header:
61
62 \noindent{\bf QCM \hfill TEST}
63
64 \vspace*{.5cm}
65 \begin{minipage}{.4\linewidth}
66 \centering\large\bf History and geography\\ Examination on Jan. 1st, 2008
67 \end{minipage}
68 \namefield{\fbox{\begin{minipage}{.5\linewidth}
69 Firstname and lastname:
70
71 \vspace*{.5cm}\dotfill
72 \vspace*{1mm}
73 \end{minipage}}}{
74
75 %%% end of the header
76
77 \begin{center}
78 \hrule\vspace{2mm}
79 \bf\Large Geography
80 \vspace{1mm}\hrule
81 \end{center}
82
83 \shufflegroup{geographie}
84 \insertgroup{geographie}
85
86 \begin{center}
87 \hrule\vspace{2mm}
88 \bf\Large History
89 \vspace{2mm}\hrule
90 \end{center}
91
92 \shufflegroup{histoire}
93 \insertgroup{histoire}
94
95 \clearpage
96
97 }
98
99 \end{document}

```

An example with a separate answer sheet

In this example, one wants the check-boxes to be put together in a separate sheet. This makes cheating more difficult, and, more importantly, it is enough to scan a single sheet per student, which makes things easier if one has to do a manual scan. In this example, the number of questions is limited: they fit into a single page, so that such a layout would not be really useful in this particular case. It is up to you to modify this example in order to use this layout with a large number of questions!

```

1 \documentclass[a4paper]{article}
2
3 \usepackage[utf8x]{inputenc}
4 \usepackage[T1]{fontenc}
5
6 \usepackage[box,completemulti,separateanswersheet]{automultiplechoice}
7 \begin{document}
8
9 \AMCrandomseed{1237893}

```

```

10
11 \def\AMCformQuestion#1{\vspace{\AMCformVSpace}\par {\sc Question #1:} } 2[2]
12
13 \element{general}{
14   \begin{question}{prez}
15     Among the following persons, which one has ever been a President of the French Republic ←
16     ?
17     \begin{choices}
18       \correctchoice{René Coty}
19       \wrongchoice{Alain Prost}
20       \wrongchoice{Marcel Proust}
21       \wrongchoice{Claude Monet}
22     \end{choices}
23   \end{question}
24 }
25
26 \element{general}{
27   \begin{questionmult}{pref}
28     Among the following cities, which ones are french prefectures?
29     \begin{choices}
30       \correctchoice{Poitiers}
31       \wrongchoice{Sainte-Menehould}
32       \correctchoice{Avignon}
33     \end{choices}
34   \end{questionmult}
35 }
36
37 \element{general}{
38   \begin{question}{nb-ue}
39     How many different states were members of the European Union in Jan. 2009?
40     \begin{choiceshoriz}[o]
41       \wrongchoice{15}
42       \wrongchoice{21}
43       \wrongchoice{25}
44       \correctchoice{27}
45       \wrongchoice{31}
46     \end{choiceshoriz}
47   \end{question}
48 }
49
50 \onecopy{5}{
51   %% beginning of the test sheet header:
52
53   \noindent{\bf QCM \hfill TEST}
54
55   \vspace*{.5cm}
56   \begin{minipage}{.4\linewidth}
57     \centering\large\bf Test\ Examination on Jan. 1st, 2008
58   \end{minipage}
59
60   \begin{center}\em
61   Durée : 10 minutes.
62
63   No documents allowed. The use of electronic calculators is forbidden.
64
65   Questions using the sign \multiSymbole{} may have
66   zero, one or several correct answers. Other questions have a single correct answer.
67
68   Negative points may be attributed to \emph{very
69   bad} answers.
70

```

```

71 \end{center}
72 \vspace{1ex}
73
74 %%% end of the header
75
76 \shufflegroup{general}
77 \insertgroup{general}
78
79 \AMCcleardoublepage ❸[3]
80
81 \AMCformBegin ❹[4]
82
83 %%% beginning of the answer sheet header
84
85 {\large\bf Answer sheet:}
86 \hfill \namefield{\fbox{ ❺[5]
87     \begin{minipage}{.5\linewidth}
88         Firstname and lastname:
89
90         \vspace*{.5cm}\dotfill
91         \vspace*{1mm}
92     \end{minipage}
93 }}
94
95 \begin{center}
96     \bf\em Answers must be given exclusively on this sheet:
97     answers given on the other sheets will be ignored.
98 \end{center}
99
100 %%% end of the answer sheet header
101
102 \AMCform ❻[6]
103
104 \clearpage
105
106 }
107
108 \end{document}

```

The following remarks should make the above example clearer :

- ❶ The `separateanswersheet` option is what allows us to do what we wanted.
- ❷ One can re-define in this manner the way the questions are identified on the answer sheet (this line is optional).
- ❸ This page break is put before the special page where the check-boxes are put together. If one does recto-verso printing, it is preferable to use `\AMCcleardoublepage` so that this page is printed apart from the others. In the case of recto printing, one can simply use `\clearpage`.
- ❹ This command marks the beginning of the answer sheet part. Its use is necessary for the appropriate treatment of the questions which appear only in that part, e.g. those generated by **AMCcode**.
- ❺ Students should normally write their name on the answer sheet!
- ❻ The LaTeX command **AMCform** writes all the check-boxes.



Warning

When one uses a separate answer sheet, letters (or digits, if one uses the option `digits`, see Section 5.4.1) are written in the check-boxes. To achieve a correct detection of the checked boxes, one has to ask students to completely fill the relevant boxes (checking by simply drawing a cross would not suffice). One also has to tune the grey level threshold (defining the proportion of black dots in a box above which that box is considered to be checked) to a value of order 0.5.

Description of the LaTeX commands

Package options

To use the `automultiplechoice` package, one uses the line

```
\usepackage[...]{automultiplechoice}
```

where the dots are replaced by a list of options separated by commas. Here are the available options:

- `lang=XX`: sets the subject language to `XX`. At present, only DE (German), ES (Spanish), FR (French), IT (Italian), JA (Japanese) and NL (Dutch) are available. Several strings added by `automultiplechoice` will be translated, such as "None of these answers are correct", see option `completemulti`.
- `box`: puts every question in a block, so that it cannot be split by a page break.
- `completemulti`: automatically adds a "None of these answers are correct" choice at the end of each multiple question. Thus, for these questions, a distinction can be made between no answer and the answer consisting in treating none of the listed answers as correct. This behaviour can be forced or cancelled for a particular question using one of the commands `\AMCcompleteMulti` or `\AMCnoCompleteMulti` inside the corresponding `questionmult` environment.
- `noshuffle`: stops the automatic shuffling of the answers for every question
- `answers`: produces the corrected version of the MCQ test, not the test sheet itself.
- `separateanswersheet`: requires that all check-boxes be put together at the end of the test sheet (usually, this option is used when one wants to have only one sheet to scan per student - see the example in Section 5.3).
- `digits`: if one uses the `separateanswersheet` option, the `digits` option requires the question to be identified with digits rather than with letters (which corresponds to the default setting). With this option, one has to take care that the number of answers does not exceed 9.
- `outsidebox`: when using `separateanswersheet`, this option asks to print letters (or digits) outside the boxes on the answer sheet.
- `insidebox`: when not using `separateanswersheet`, this options asks to print letters (or digits) inside the boxes to be filled by the students.
- `catalog`: use this option to make a catalog of your questions to be used to compose future exams. No need to use `\onecopy` with this layout.
- `postcorrect`: use this option if you want to give the correct answers after scans analysis, from a teacher completed answer sheet - see Section 7.2 for details.
- `automarks`: in `separateanswersheet` mode, use this option if you want to cancel marks printing on the subject pages. They will only be print on the answer sheet pages. To change the way pages numbers are print on the subject pages, redefine the `\AMCsubjectPageTag` command:

```
\renewcommand\AMCsubjectPageTag{%
  \fbox{\texttt{\the\AMCid@etud:\thepage}}%
}
```



Warning

Only use `automarks` option if no data are to be collected on the subjects pages, as AMC won't be able to process these pages.

Description of a copy

The LaTeX source code describing the content of the test sheet has to be included in a call to the command `\onecopy`, the first argument being the number of distinct realizations to be produced, and the second argument being the code used to generate a realization.

```
\onecopy{50}{ ... }
```

If you have downloaded the LaTeX package `environ` *before* `automultiplechoice`, an alternative syntax is available, using the **examcopy** environment, where the number of realizations is an option (default is 5).

```
\begin{examcopy}[50]
...
\end{examcopy}
```

Note

The `environ` package is not available in the TeX Live 2007 distribution, which is still used in Ubuntu distributions up to version 9.10 (Karmic Koala).

Questions and answers

For simple questions (a single correct answer), one uses the following model:

```
\begin{question}{identifiant}
  Here is the question...
  \begin{choices}
    \correctchoice{The correct answer}
    \wrongchoice{A wrong answer}
    \wrongchoice{Another wrong answer}
  \end{choices}
\end{question}
```



Important

One must use a different identifier for every question. An identifier can be made of digits, letters, and simple characters (but do not use e.g. braces or brackets). Don't end your question identifier with a dot followed by digits, as this syntax is reserved to codes input.

Note

The maximum number of answers for a given question is limited to 199.

To keep the original order of the answers and prevent shuffling for this specific question, one can use the `o` option of the `choices` environment, replacing line 3 by the following:

```
\begin{choices}[o]
```

To put the answers on two columns, one can use the `multicol` package: load it in the preamble (just after the reference to the package `automultiplechoice` for instance) with

```
\usepackage{multicol}
```

and include the `choices` environment inside a `multicols` environment in the following manner:

```
\begin{multicols}{2}
  \begin{choices}
    \correctchoice{The correct answer}
    \wrongchoice{A wrong answer}
    \wrongchoice{Another wrong answer}
  \end{choices}
\end{multicols}
```

For even shorter answers, one can require questions to be printed following one another, using the `choiceshoriz` environment instead of `choices`.

Multiple questions (those for which no, one, or several answers can be correct) use the `questionmult` environment instead of `question`.

When the answer to the question is not supposed to be taken into account in the grading, one uses the `\QuestionIndicative` command, as in the following example:

```
\begin{question}{difficulty}\QuestionIndicative
  \scoring{auto=0,v=-1,e=-2}
  Did you find this class easy or difficult? Please answer on a scale from 0 (very ←
    difficult) to 5
  (very easy).
  \begin{choiceshoriz}[o]
    \correctchoice{0}
    \correctchoice{1}
    \correctchoice{2}
    \correctchoice{3}
    \correctchoice{4}
    \correctchoice{5}
  \end{choiceshoriz}
\end{question}
```

Putting answers on multiple columns

To put answers on several columns (and thus save space), one can embed the `choices` environment in a `multicols` environment, using the LaTeX package `multicol`.

If, moreover, the answers do not fit into a single line, an answer might be split over several columns, which might be a little puzzling for the reader. The `\AMCBoxedAnswers` command was defined in order to prevent this phenomenon, by embedding each answer into a box. Here is an example of use:

```
\begin{question}{two columns}
  What is a bird ?
  \begin{multicols}{2}\AMCBoxedAnswers
    \begin{choices}
      \correctchoice{It is an animal with wings, laying eggs. There are birds with all ←
        sorts of colors.}
      \wrongchoice{It is a large piece of furniture, made of wood, and used most of the ←
        time to store
        household linen}
      \wrongchoice{It is a steam machine devised to seal cans at high speed.}
    \end{choices}
  \end{multicols}
\end{question}
```

Let us note that it is also possible to parametrize the vertical space between two answer blocks, thanks to the dimension `AMCinterBrep`:

```
\AMCinterBrep=.5ex
```

Groups of questions

Putting questions into groups allows one to shuffle questions inside these groups, in a different way for each realization of the test. Every group of questions must have a name formed solely with plain letters.

One can put questions in a group one by one, as in the following example.

```
\element{mygroup}{
  \begin{question}{easy}
    So, how much is one plus one?
    \begin{choiceshoriz}
      \correctchoice{two}
      \wrongchoice{zero}
      \wrongchoice{three}
    \end{choiceshoriz}
  \end{question}
}
```



Important

The formation of the group, using the **element** commands, must be made only once: thus, these commands have to be used *before* the **onecopy** command, which repeats some actions for every realization.

Once a group is formed, it is possible to shuffle questions inside this group using the **shufflegroup** command. For instance

```
\shufflegroup{mygroup}
```

Finally, the group content can be output to the test sheet using command **insertgroup**, as in

```
\insertgroup{mygroup}
```

Note

These last two commands (**shufflegroup** and **insertgroup**) are to be used inside the argument of the command **onecopy**, so that shuffling is performed before each realization.

Groups of questions can be manipulated more precisely thanks to the following commands:

- `\insertgroup[n]{mygroup}` (using optional parameter n) only inserts the n first elements from the group.
- `\cleargroup{mygroup}` clears all group content.
- `\copygroup{groupA}{groupB}` copies all the elements from group *groupA* to the end of group *groupB*. With an optional argument n , only the n first elements will be copied: `\copygroup[n]{groupA}{groupB}`.

With these commands, you can for exemple make a exam taking 4 questions from group GA at random, 5 questions from group GB at random, and all the questions from group GO, shuffling all these questions, with the following code (to be used inside the argument of the command **onecopy**):

```
\cleargroup{all}
\shufflegroup{GA}\copygroup[4]{GA}{all}
\shufflegroup{GB}\copygroup[5]{GB}{all}
\copygroup{GO}{all}
\shufflegroup{all}
\insertgroup{all}
```


Page size, margins

The **automultiplechoice** LaTeX package uses **geometry** to set the margins and page layout. You can overwrite its settings using the **\geometry** command just before the `\begin{document}` - see the geometry package documentation for details. The values initially set by AMC are:

```
\geometry{hmargin=3cm,headheight=2cm,headsep=.3cm,footskip=1cm,top=3.5cm,bottom=2.5cm}
```

When reducing the margins to gain some space, keep in mind that:

- The four corner marks must be printed entirely (they could disappear due to the printer margins).
- The four corner marks must be entirely visible on the scans (if they are too close from the border and the paper moved a little or turned a little during scanning, this could not be the case).

It is also possible to set the paper size as an option to add to the list given as argument to the **\geometry** command. Available values include `a4paper`, `a5paper`, `a6paper`, `b4paper`, `b5paper`, `ansipaper`, `ansicpaper`, `ansidpaper`, `letterpaper`, `executivepaper`, `legalpaper`.

For small paper sizes, it may also be interesting to change the position of the human readable sheet IDs (like +1/1/53+) in the header. This can be done using the **\AMCidsPosition** command, in the form

```
\AMCidsPosition{pos=p,width=w,height=h}
```

where p can be `none`, `top` or `side`, and the dimensions w and h refers to the (invisible) box containing the ID. The default values are

```
\AMCidsPosition{pos=side,width=4cm,height=3ex}
```

Let us end with an example for A5 paper sheets:

```
\geometry{a5paper,hmargin=1.6cm,top=2.5cm}
\AMCidsPosition{pos=top}
```

Check-box presentation style

The **\AMCboxDimensions** command allows one to modify one or several dimensions of the check-boxes.

- `shape` is the shape of the boxes. Use the value `square` to get squares or rectangles, and `oval` to get circles or ovals. Note that the LaTeX package `tikz` must be loaded for `oval` to work.
- `width` is the width of the box;
- `height` is the height of the box;
- `size` is the size (width and height) of the box;
- `rule` is the thickness of the boundary of the box;
- `down` controls by how much boxes are pushed down.

To obtain smaller boxes, one can e.g. use the command

```
\AMCboxDimensions{size=1.7ex,down=.2ex}
```

When using `separateanswersheet` package option, you can also customize the boxes labelling. The default behaviour is to use uppercase alphabetical labelling, or arabic numbering if the `digits` package option is used. To use your own labelling, one has to redefine the **\AMCchoiceLabel** command which takes as argument the name of the counter used to number the choices. For example, the following code will ask for lowercase letters to label the boxes:

```
\def\AMCchoiceLabel#1{\alph{#1}}
```

As an other example, when using `arabxetex` package, the following code may be useful:

```
\def\AMCchoiceLabel#1{\textLR{\Alph{#1}}}
```

One can also change the style of the boxes labels redefining the `\AMCchoiceLabelFormat` command, as in the following example (here we need bold labels):

```
\def\AMCchoiceLabelFormat#1{\textbf{#1}}
```

Also note that the color of the boxes to be ticked by the students can be set using

```
\AMCboxColor{red}
```

replacing `red` by the desired color (that must be recognized by the `xcolor` package). For example, you can get a light green with

```
\definecolor{amcboxcolor}{rgb}{0.67,0.88,0.5}
\AMCboxColor{amcboxcolor}
```

Questions presentation style

The way each question is presented can be modified by redefining the LaTeX command `AMCbeginQuestion`, whose default definition is the following:

```
\def\AMCbeginQuestion#1#2{\par\noindent{\bf Question #1} #2\hspace*{1em}}
```

The first parameter transmitted to this command is the number of the question to be displayed. The second parameter contains `\multiSymbole` in the case of a multiple question, and is void in all other cases. The `\multiSymbole` command too can be modified: its goal is to distinguish multiple questions from the others. By default, it displays a club.

```
\def\multiSymbole{$\clubsuit$}
```

Finally, one can modify the number of the next question with the `\AMCnumero` command. At the beginning of each realization of the test, a call to

```
\AMCnumero{1}
```

is performed, but this command can be used at any place.

The display of answers can be modified in the same fashion, if one uses the `choicescustom` environment instead of `choices` or `choiceshoriz`, redefining the three following LaTeX macros:

```
\def\AMCbeginAnswer{}
\def\AMCendAnswer{}
\def\AMCanswer#1#2{#1 #2}
```

One can also change spacing redefining the following dimensions (here are the default values):

```
\AMCinterIrep=0pt
\AMCinterBrep=.5ex
\AMCinterIquest=0pt
\AMCinterBquest=3ex
```

These dimensions are the vertical space between questions (`quest`) or answers (`rep`), in boxed mode (B, with `\AMCBoxedAnswers` or `box` package option) or standard mode (I).

Separate answer sheet presentation style

It is also possible to modify the layout of the separate answer sheet produced with the `separateanswersheet` option (see section Section 5.3).

1. If one only wants to modify the horizontal spacing between two check-boxes or the vertical spacing between two questions, one just has to redefine the following dimensions:

```
\AMCformHSpace=.3em
\AMCformVSpace=1.2ex
```

2. For a deeper modification of the display settings, one can also redefine the commands that are used at the beginning of each question and for each answer (here follows the default definitions):

```
\def\AMCformQuestion#1{\vspace{\AMCformVSpace}\par{\bf Question #1:}}
\def\AMCformAnswer#1{\hspace{\AMCformHSpace} #1}
```

These definitions have to be inserted just after `\begin{document}` in the LaTeX file.

Code acquisition

Code acquisition can be easily performed thanks to the LaTeX command **AMCcode**, for instance to allow each student to enter her/his student number by herself/himself on the answer sheet. The two arguments of this command are a code/question identifier, and the number of digits to be used by the code. One can e.g. use the following header

```
{\setlength{\parindent}{0pt}\hspace*{\fill}\AMCcode{etu}{8}\hspace*{\fill}
\begin{minipage}[b]{6.5cm}
$\longleftarrow{\hspace{0pt plus 1cm}}$ please encode your student number below,
and write your first and last names below.

\vspace{3ex}

\hfill\namefield{\fbox{
\begin{minipage}{.9\linewidth}
  Firstname and lastname:

  \vspace*{.5cm}\dotfill

  \vspace*{.5cm}\dotfill
  \vspace*{1mm}
\end{minipage}
}}\hfill\vspace{5ex}\end{minipage}\hspace*{\fill}
}
```

If the `separateanswersheet` option is used, the **AMCcode** command has to be placed after the **AMCformBegin** command.

Note that the codes rendering from **AMCcode** can be adapted modifying the lengths **AMCcodeHspace**, **AMCcodeVspace** and **AMCcodeBoxSep**, representing the horizontal and vertical amount of space between boxes, and the amount of space between boxes and labels (when the labels are written outside the boxes). Default values are set with the following commands:

```
\AMCcodeHspace=.5em
\AMCcodeVspace=.5em
\AMCcodeBoxSep=.1em
```

Note that an horizontal version **AMCcodeH** is also available, specially for the case of small number of digits.

Open questions

One can sometimes require some open questions to be added to the subject. One way to do so is to reserve boxes use to the teacher for these questions. After the exam, the teacher reads the completed answer sheets and tick the boxes according to the answers written by the students for open questions. He will then scan the sheets and let AMC correct the multiple choice questions and integrate the open questions scores into the marks.

```
\begin{question}{open}
  Define \emph{inflation}.
  \AMCOpen{lines=5}{\wrongchoice[W]{w}\scoring{0}\wrongchoice[P]{p}\scoring{1}\ ←
    correctchoice[C]{c}\scoring{2}}
\end{question}
```

In this example, the teacher will get three boxes. If the first (labelled W for wrong) is ticked, the student will get 0 points. If the second (labelled P for partial) is ticked, the student will get 1 point. If the third (labelled C for correct) is ticked, the student will get 2 points.

The first argument to **\AMCOpen** is a comma separated list of options. The available options are:

lineup=bool if `true`, the answering area and the scoring boxes will be on the same line. If `false` (this is default), the answering area is enclosed in a frame and placed below the scoring boxes.

lines=num sets the number of lines for the answer. Default value is 1.

lineheight=dim sets the height of each line. Default value is 1cm.

dots=bool if `true` (default), each line will be realized by a line of dots.

contentcommand=cmdname Use this option if you want to customize the content of the answer area. You will have to define a `\cmdname` command that has to produce the content.

hspace=dim sets the space between boxes in the marking area.

backgroundcol=color sets the background color of the marking area.

foregroundcol=color sets the foreground color of the marking area.

scan=bool if `false`, the boxes are not scanned (this can be useful if you plan to use manual data capture only to mark this question, and don't want to take into account the students drawings on the boxes). Defaults to `true`.

annotate=bool if `false` (default value), the boxes from this question won't be annotated when annotating the answer sheets (only the score will be written).

question=text sets a short text that helps the examiner to identify the question. This text will be written before the scoring boxes, only if a separate answer sheet is used.

answer=text sets a short text that will be written in the answering area for corrected answer sheets.

width=dim sets the width of the frame enclosing the answering area when `lineup=false`. Default value is `.95\linewidth`.

framerule=dim sets the line width for the frame enclosing the answering area.

framerulecol=color sets the frame color for the answering area.

boxmargin=dim sets the margin around the scoring boxes.

boxframerule=dim sets the line width for the frame around the scoring boxes.

boxframerulecol=color sets the color of the frame around the scoring boxes.

You can set other default values (for the whole exam) for all these parameters using the command **\AMCopenOpts**, like

```
\AMCopenOpts{boxframerule=2pt,boxframerulecol=red}
```

Moreover, a little text can be written in the marking area to tell the students not to tick these boxes, redefining the `\AMCotextReserved` command, as:

```
\def\AMCotextReserved{\emph{Reserved}}
```

When using separate answer sheets, the text added to each open question can also be defined, as:

```
\def\AMCotextGoto{\par{\bf\emph{Please write the answer on the separate answer sheet.}}}
```

Choice of shuffling parameters

One can modify the seed of the random number generator used to produce the shuffle, thanks to the following command (to be used just at the beginning of the document, and in any case outside the `\onecopy` command):

```
\AMCrandomseed{1527384}
```

If the value that is assigned (to be chosen between 1 and 4194303) is modified, then the shuffling will differ. Of course, one must not modify this value after the test sheets have been printed.

Using sectioning and separate answer sheet

For sectioning to be also visible in the separate answer sheet (if any), use `\AMCsection` and `\AMCsubsection` instead of `\section` and `\subsection`.

Using references inside the test sheets

Using LaTeX commands `\label`, `\ref` and `\pageref` within questions or answers is problematic since they will be called with the same arguments for every realization of the test, which will alter the numbering of references. To solve this problem, one should use instead the commands `\AMClabel`, `\AMCref` and `\AMCpageref`: they add the realization number to their argument before transmitting it to `\label`, `\ref` and `\pageref`.

One also has to reset counters to zero at the beginning of each realization. For instance, if one wants to include references to pictures that are put together in a separate page, one might write something like

```
\element{animals}{
  \begin{figure}[p]
    \centering
    \includegraphics[width=.6\linewidth]{tiger}
    \caption{An animal}
    \AMClabel{tiger}
  \end{figure}

  \begin{question}{tiger}
    Which is the animal on figure~\AMCref{tiger} page~\AMCpageref{tiger}?
    \begin{choices}
      \correctchoice{A tiger}
      \wrongchoice{A giraffe}
      \wrongchoice{An elephant}
      \wrongchoice{A cat}
    \end{choices}
  \end{question}
}
```

and it is then important to add, just after the command `\onecopy` the line

```
\setcounter{figure}{0}
```

so that the numbering of figures starts at 1 for every realization. Without that last command, the numbering of figures would go on from one realization to the next one, which is clearly not desirable.

Customizing some texts inserted by AMC

Use `\AMCtext` for the following customizations:

- `\AMCtext{none}{sentence}` replaces « None of these answers are correct. » (the English default text) with the given *sentence* when using option `completemulti`.
- `\AMCtext{corrected}{title}` replaces « Corrected » (the English default text) with the given *title* on the corrected answer sheet.
- `\AMCtext{catalog}{title}` replaces « Catalog » (the English default text) with the given *title* on the questions catalog produced thanks to option `catalog`.
- You can also consider using commands like the following ones (here the second argument is set to the default English value):

```
\AMCtext{draft}{DRAFT}
\AMCtext{message}{For your examination, preferably print documents
                  compiled from auto-multiple-choice.}
```

Options for AMC

You can add in the source file header (the first lines that begin with a '%') some options to be passed to AMC:

%%AMC:preprocess_command=commandname tells AMC to run the *commandname* command before calling LaTeX to process the source file. This command will be run inside the project directory, and the name of a source file copy will be passed as an argument. There is no problem for *commandname* to change this file content, because this is only a copy.

%%AMC:latex_engine=engine tells AMC to set the LaTeX engine to use for this file, regardless to the one entered as a preference by the user.

Mathematical questions with randomized statements

Using package fp

Thanks to the LaTeX package `fp`, whose documentation is available at <http://mirrors.ctan.org/macros/latex/contrib/fp/README>, and which can be downloaded with the command

```
\usepackage{fp}
```

added before that corresponding to `automultiplechoice`, one can create exercises with randomized numerical data. Let us start with a simple example.

```
\begin{question}{addition}
  \FPeval\VQa{trunc(1+random*8,0)} ❶[1]
  \FPeval\VQb{trunc(4+random*5,0)}
  \FPeval\VQsomme{clip(VQa+VQb)} ❷[2]
  \FPeval\VQnonA{clip(VQa+VQb-1)} ❸[3]
  \FPeval\VQnonB{clip(VQa*VQb)}
  \FPeval\VQnonC{clip(VQa-VQb)}

  What is the sum of \VQa{} et \VQb{} ?
  \begin{choiceshoriz}
    \correctchoice{\VQsomme}
    \wrongchoice{\VQnonA}
    \wrongchoice{\VQnonB}
    \wrongchoice{\VQnonC}
  \end{choiceshoriz}
\end{question}
```

The `\FPeval` command is used to perform computations:

- ❶ Since **random** returns a real number in the interval $[0,1]$, this command sets VQa to a random integer value between 1 et 8. The next line sets VQb to a random integer value between 4 et 8.
- ❷ Putting the correct answer in the variable $VQsomme$.
- ❸ Putting wrong answers in variables $VQnonA$, $VQnonB$ et $VQnonC...$

Variable names beginning with VQ have been chosen so as to avoid interference with other LaTeX commands.

Choosing an interval

The **automultiplechoice** package moreover defines a `\AMCIntervals` command which makes this kind of construction simpler, as illustrated in the next example:

```
\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 the exponential
  distribution with
  respective parameters  $\text{\VQa}$  et  $\text{\VQb}$ .
  To which interval does the probability  $\mathbb{P}[X<Y]$  belong ?

  \begin{multicols}{5}
    \begin{choices}[o]
      \AMCIntervals{\VQr}{0}{1}{0.1} ❶[1]
    \end{choices}
  \end{multicols}
\end{question}
```

- ❶ This lines inserts ten answers corresponding to the intervals $[0,0.1[$ $[0.1,0.2[$... $[0.9,1[$, while indicating that the correct interval is the one containing VQr . The arguments of the `\AMCIntervals` command are the following:

1. The correct answer,
2. The left point of the first interval,
3. The right point of the last interval,
4. The length of each interval.

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

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

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

Coding the result

The students can also be asked to code the numerical answer, using the `\AMCnumericChoices` command, as in the following example:

```
\begin{questionmultx}{sqrt}
  \FPeval{VQa}{trunc(5+random*15,0)}
  \FPeval{VQs}{VQa^0.5}
  Compute  $\sqrt{\text{\VQa}}$  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 symbol for multiple questions.

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 *num* 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.

nozero=bool if `true`, the choice 0 is removed for all digits. May be useful when using `\AMCnumericChoices` to enter small (<10) positive values.

sign=bool requests (or not) a signed value (defaults 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.

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

scoring=bool if `true`, a scoring strategy is given to AMC for this question. Defaults to `true`.

scoreexact=num gives the score for an exact answer (defaults to 2).

exact=num 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=num gives the score for an *approximative* answer (defaults to 1).

approx=num 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).

You can set other default values (for the whole exam) for all these parameters using the command `\AMCnumericOpts`, like


```
\AMCnumericOpts{scoreexact=3,borderwidth=2pt}
```

Moreover, 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 **\AMCnxtGoto** command, as:

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

Using package pgf/tikz

LaTeX package `pgf/tikz` (see <http://www.ctan.org/tex-archive/graphics/pgf/base>) provides mathematical functions that can be loaded with

```
\usepackage{tikz}
```

First of all, you must set the random seed to be sure to get the same result each time latex is run to compile the subject:

```
\pgfmathsetseed{2056}
```

A simple computation

Here is an exemple with a simple computation:

```
\begin{question}{inverse}
  \pgfmathrandominteger{\x}{1}{50}

  How much is the reciprocal of  $x=\x$ ?

  \begin{choices}
    \correctchoice{\pgfmathparse{1/\x}\pgfmathresult }
    \wrongchoice{\pgfmathparse{1/(\x +1)}}\pgfmathresult}
    \wrongchoice{\pgfmathparse{cos(\x)} }\pgfmathresult}
    \wrongchoice{\pgfmathparse{\x^(-0.5)}\pgfmathresult}
  \end{choices}
\end{question}
```

Here, the command **\pgfmathparse** makes the computation, and **\pgfmathresult** outputs the result.

Output formatting is also available with the command **\pgfmathprintnumber**, as in the following exemple (three digits after decimal point, and use of the comma as a decimal point).

```
\begin{question}{inverse3}
  \pgfmathrandominteger{\x}{1}{50}
  \pgfkeys{/pgf/number format/.cd,fixed,fixed zerofill,precision=3,use comma}
  How much is the reciprocal of  $x=\pgfmathprintnumber{\x}$ ?

  \begin{choices}
    \correctchoice{\pgfmathparse{1/\x}\pgfmathprintnumber{\pgfmathresult}}
    \wrongchoice{\pgfmathparse{1/(\x +1)}} \pgfmathprintnumber{\pgfmathresult}}
    \wrongchoice{\pgfmathparse{cos(\x)} }\pgfmathprintnumber{\pgfmathresult}}
    \wrongchoice{\pgfmathparse{\x^(-0.5)} }\pgfmathprintnumber{\pgfmathresult}}
  \end{choices}
\end{question}
```

Note

You can also use the **\AMCIntervals** and **\AMCnumericChoices** commands (see Section 5.6.1.1 and Section 5.6.1.2)

Graphics

The `tkiz` package also allows to make (random or not) graphs.

```
\begin{questionmult}{graph}
  Let us consider the three functions which graphs are plotted below:
  \pgfmathrandominteger{\a}{2}{4}
  \begin{center}
    \begin{tikzpicture}[domain=0:4]
      \draw[very thin,color=gray] (-0.1,-4.1) grid (3.9,3.9);
      \draw[->] (-0.2,0) -- (4.2,0) node[right] {$x$};
      \draw[->] (0,-4.2) -- (0,4.2) node[above] {$f(x)$};
      \draw[color=red] plot (\x,{(1+\a/4)*\x-\a}) node[right] {$f_{1}(x)$};
      \draw[color=blue] plot (\x,{\a*sin(\x r)}) node[right] {$f_{2}(x)$};
      \draw[color=orange] plot (\x,{\a*cos(\x r)}) node[right] {$f_{3}(x)$};
    \end{tikzpicture}
  \end{center}
  Then:
  \begin{choices}
    \pgfmathrandominteger{\x0}{2}{4}
    \correctchoice{$f_{2}(\x0)$=\pgfmathparse{\a*sin(\x0 r)} \pgfmathprintnumber{\x0} \leftarrow \pgfmathresult}{}
    \correctchoice{$f_{3}(\x0)$=\pgfmathparse{\a*cos(\x0 r)} \pgfmathprintnumber{\x0} \leftarrow \pgfmathresult}{}
    \wrongchoice{La fonction $f_{1}(x)$ est une fonction linéaire.}
  \end{choices}
\end{questionmult}
```

To make pretty graphs, package `pgfplots` can also be useful.

With `pgfmath`, precision is limited, so that a "Arithmetic overflow" error can be encountered. Packages `tikz` and `pgfplots` allows to overcome this problem, using **gnuplot** as a backend. You need to install **gnuplot** on your system, and use LaTeX option `--shell-escape`. To this purpose, go to AMC preferences window, and set the LaTeX engine for your project to `"pdflatex --shell-escape"` (without the quotes).

Using LuaLaTeX

LUA language can be used inside LaTeX documents thanks to the **lualatex** command. If you uses it, your document needs to be UTF-8 encoded, and you must not load the `inputenc` package. See <http://www.luatex.org/documentation.html> for some information.

LUA commands are to be given as a `\directlua` argument. The most useful LUA function is **tex.print**, which will output results back to LaTeX.

Once again, if you use random numbers, allways fix the random seed to get the same results across differents compilations:

```
\directlua{math.randomseed (2048)}
```

Here is a very simple sample source file:

```
\documentclass[a4paper]{article}
%\usepackage[utf8x]{inputenc}
\usepackage[T1]{fontenc}
\usepackage[box,completemulti]{automultiplechoice}
\begin{document}

\onecopy{10}{

%% head

\noindent{\bf QCM \hfill TEST}
```

```

\vspace*{.5cm}
\begin{minipage}{.4\linewidth}
  \centering\large\bf LuaLaTeX sample exam
\end{minipage}
\namefield{\fbox{\begin{minipage}{.5\linewidth}
Name :

\vspace*{.5cm}\dotfill
\vspace*{1mm}
\end{minipage}}}{}}

%%

\directlua{math.randomseed (2048)}
\directlua{a=math.random() }

\begin{question}{square-root}
  How much is the square root of \directlua{tex.print(a)}?

  \begin{choices}
    \correctchoice{\directlua{tex.print(math.sqrt(a))}}
    \wrongchoice{\directlua{tex.print(math.sqrt(2*a))}}
    \wrongchoice{\directlua{tex.print(math.sqrt(a*1.001))}}
  \end{choices}
\end{question}

\clearpage
}

\end{document}

```

Output formatting can be obtained using lua functions, or with the `siunitx` package.

Note

You can also use the `\AMCIntervals` and `\AMCNumericChoices` commands (see Section 5.6.1.1 and Section 5.6.1.2)

Usage - Graphical interface

We shall describe here a usage example with the graphic interface from the conception of the multiple-choice test until the edition of students' scores.

Creating a new project and subject

Let's open the graphic interface. This can be done ordinarily by selecting Applications → Education → Auto Multiple Choice in the general menu of Gnome (or its equivalent in KDE or other), but the command **auto-multiple-choice** can be used directly.

Let's now create a new project, thanks to Project → New. A window opens and allows us to see existing project names (if any) and choose a name for our new project (made with simple characters; « test » will be OK for our short test), which we write in the field Project name. Then we push the New project button.

Now we must choose a LaTeX file as a source for the multiple-choice. Several possibilities are shown:

- **model**: this choice allows to choose from models shipped with AMC an exam to customize later.
 - **file**: this choice allows to choose a LaTeX file already prepared for this exam. Somebody may have prepared the exam for you, or you can have prepared the exam outside AMC, using your favorite LaTeX editor.
-

- empty: using this choice, an empty LaTeX file will be created. You have to edit it to compose the exam from zero.
- archive: use this choice if you have a zip or tgz archive containing the exam definition (LaTeX source file, along with image files, parameters file for example). This archive can be made by an external software. It can also be a backup of one of your AMC projects.

For our test, let us choose model. The next window presents the models: choose for example Simple example from [EN] Documentation group. Now we can edit it to modify the shape of the document or the questions, thanks to the Edit LaTeX file button which launches the default editor.

Preparing the subject

Preparing a project is done in two steps. First we must make the reference documents from the LaTeX source file. This can be done by clicking the Update documents. The following documents are produced:

- The *question*. This file can be printed to distribute its pages to students (see below).
- The *solution*. We can check that the chosen responses there are the good ones. It is also made to be distributed to students.

When produced, those documents can be viewed (and possibly printed) from the corresponding buttons.

Now we can begin the last step of the preparation: analyzing the layout. It can be launched with the button Layout detection. This analysis detects, in every page of the subject, the exact position of every element which must be analyzed in the students' copies.

To check whether the layouts have been correctly detected, we can use the button Check layouts. A short insight allows to check that red checkboxes are correctly located over the boxes of the subject.

Printing and exam

Two alternative workflows can be considered:

- For the most robust, create as many exam sheets as necessary for all your students, with different sheets numbers, and print them all. Each page can be identified by its numbers and boxes at the top, so that you can scan several times the same completed answer sheet page carefree.
- Secondly, you can print a few subjects (or only one if you want), and photocopy them to get one subject for all students. Questions shuffling will be less efficient, and if you give several times a scan of the same page, AMC won't be able to know about it and will create an unwanted duplicate.



Warning

To use this second workflow using photocopies, there must be only one page for students to write on (using a separate answer sheet can help you for this). If not, you won't be able to continue with AMC! Indeed, it would be impossible for AMC to make the link between two pages from the same student.

When the preparation is over, we can print the subject, and distribute it to the students... In simple cases, we can directly print from the viewer (after clicking the line Subject in the list of work documents). When it is better to print the copies separately (for example if copies contain multiple pages and when the printer allows to staple them together), we shall rather use the button Print copies after calculating the layout.



Important

When the subject is printed and distributed, we may no more modify the *work documents* because they must remain identical to distributed copies.

Reading the copies

Now we shall describe the input from students' copies, which can be done automatically and/or manually.

Let's move to the Data capture tab of the graphical interface.

Automated input

For automatic recognition of the checked boxes in the students' pages, they must be previously digitalized. I use a copier/scanner which does it automatically (all the pages in a bundle without interaction with me), with the following settings: 300 dpi, OCR mode (for the characters' recognition, black and white without greyscale - but the scanner does not process any character recognition), each scan delivered as a single TIFF file per page.

Note

To analyze the scans, we must have them in one or several image files (TIFF, JPG, PNG, etc.). Vector graphics formats (PDF, PS or EPS) are also suitable: scans will then be converted into PNG by AMC before analysis.

Note

When giving scans for automated data capture the first time, you will have to tell AMC which method you used: either different papers printed, or photocopied papers (see [Printing and exam](#)).

Then we select this set of scan files in the dialog opened by the button Automated of the section Data capture after examination, then we validate with the OK button. AMC begins with Optical Mark Recognition to detect the position of the four circle corner marks on the scans, position the boxes, and detects the amount of black pixels in each box.

The result of the analysis of each page is indicated in the lists of the section Diagnosis:

- The value *MSD* (mean square deviation) is an indication of the good framing of the marks (the four black dots surrounding each copy). When it is too great, the framing must be checked (right click on the page's line then choose page to view the scanned page and the boxes as they were detected).
- The value *sensitivity* is an indicator of proximity of the filling of the boxes with the threshold. If it is too great (from 8 to its max value 10), we must check whether the boxes recognized as checked are the good ones (a right click on the page's line then choose zoom to view the set of boxes in the copy, verify whether the detection worked correctly, and correct it if needed drag-and-dropping the boxes images).

Manual input

If we cannot use easily the scanner, or if, for a few copies, the automated input did not work as expected, we can manage the input manually. To do so, let's open the right window thanks to the button Manual of the section Input of the copies after exam. In that window, we can input the boxes which have been checked ourselves (by clicking them) on the wanted pages.

Note

Every manual input will overwrite results eventually coming from a previous or posterior automated input for the same page.

Correction

In the Marking tab of the graphic interface, the part Marking allows us to deduce the scores of the students from the inputs, but also to read the codes written by the students (see Section [5.4.10](#)).

Process

The computation of the scores is launched with the button Mark, but we must previously make the following choice:

- If we check the box Update marking scale, the scoring strategy will be first extracted from the LaTeX source file. This allows to try many strategies at the end of the correction process. This action also updates which answers are specified as correct or as wrong. Hence, potential mistakes in the answers can be easily fixed after the exam. The method to specify the strategy in the LaTeX file will be explained in the section [Scoring strategy](#) (a default scoring strategy is used when no indication is given).

When we click the button Mark, the correction is made (this can take some time if we also asked for the reading of the scale).

Scoring strategy

The strategy used to score the copies is indicated in the LaTeX source file, with the command **scoring**. It can be used in an environment **question** or **questionmult**, to set it for every response, but also in the environment **choices**, to give scaling indications about a single response. The argument of the LaTeX command scoring is made of indications like *parameter=value*, separated by comas. The usable parameters are the following (the table shows also in which context those parameters can be used):

parameter	simple		multiple		value
	Q	A	Q	A	
e	●		●		The score given when responses are incoherent: many boxes checked for a simple question, or, for a multiple question, the box "none of the responses are correct" checked while another box is also checked.
v	●		●		The score given in case of no response (no box is checked).
d			●		An offset, i.e. a value added to every score not relevant of parameters e and v.
p			●		The bottom score. If the calculation of the score in that question yields a value below the bottom value, the score is set to the bottom value.
b	●		●	●	Score for a good response to a question.
m	●		●	●	Score for a bad response to a question.
		●			Without parameter name (syntax: <code>\scoring{2}</code>), this indicates the score to give if the student has checked <i>this</i> response.
auto	●				With this parameter, the value of the response numbered <i>i</i> will be <i>auto</i> + <i>i</i> -1. This option is mainly used with <code>\QuestionIndicative</code> (see section Questions and answers).
mz	●		●		This parameter is used for a "maximum or zero" scoring: if all the answers are correct, the score is <i>mz</i> . If not, the score is zero.
haut			●		When you give this parameter a value <i>n</i> , the score given for a perfect response will be <i>n</i> , and one point will be withdrawn for each error.
MAX	●		●		Gives the maximal value given for the question (for a "question scored 5", one can write MAX=5). To be used only when it is not the same value as when one replies every good response.
formula	●		●		Gives the score to be given for the question, often using a formula that uses some variables (see Section 6.5.3), without taking b and m values into account.
set.xxx	●	●	●	●	Gives a particular value to the variable named <i>xxx</i> , that will be available in a formula. In an answer context, the value is associated to the variable only if the box is ticked.
default.xxx	●		●		Gives a value to the variable <i>xxx</i> in the case when no ticked boxes gave a value to <i>xxx</i> .
requires.xxx	●		●		Tells that the variable <i>xxx</i> has to be defined, unless the data is told <i>incoherent</i> and the question scored with the value of e.

The default scale for a simple question is $e=0, v=0, b=1, m=0$, which gives one point for a good response and no point in the other cases. The default scaling for a multiple question is $e=0, v=0, b=1, m=0, p=-100, d=0$, which gives a point for every checked box, either good or not (good box checked or wrong box not checked).

The LaTeX command `\scoring` can also be used outside question definitions, for whole examination parameters:

- `SUF= x` gives a total number of points sufficient to get the maximal mark. For example, with 10 for the maximal mark and parameter `SUF=8`, a student getting a total of 6 points will get mark $6/8 \cdot 10 = 7.5$, whatever the value of the total number of points for a perfect answer sheet.
- `allowempty= x` allows the student to leave x questions unanswered. When summing up questions scores, x unanswered questions will be cancelled.

Using all of these parameters in combination allows to define many kinds of scoring strategies, as in the following example:

```

1 \documentclass{article}
2
3 \usepackage[utf8x]{inputenc}
4 \usepackage[T1]{fontenc}
5
6 \usepackage[box,completemulti]{automultiplechoice}
7
8 \begin{document}
9
10 \element{qqs}{
11 \begin{question}{good choice}
12   How many points would you like for this question?
13   \begin{choices}
14     \correctchoice{Maximum: 10}\scoring{10}
15     \wrongchoice{Only 5}\scoring{5}
16     \wrongchoice{Two will be enough}\scoring{2}
17     \wrongchoice{None, thanks}\scoring{0}
18   \end{choices}
19 \end{question}
20 }
21
22 \element{qqs}{
23 \begin{questionmult}{added}
24   Get free points checking the following boxes:
25   \begin{choices}
26     \correctchoice{2 points}\scoring{b=2}
27     \wrongchoice{One negative point!}\scoring{b=0,m=-1}
28     \correctchoice{3 points}\scoring{b=3}
29     \correctchoice{1 point}
30     \correctchoice{Half point}\scoring{b=0.5}
31   \end{choices}
32 \end{questionmult}
33 }
34
35 \element{qqs}{
36 \begin{questionmult}{3 or zero}\scoring{mz=3}
37   Only a perfect response will be scored 3 points - otherwise, null score.
38   \begin{choices}
39     \wrongchoice{Wrong}
40     \wrongchoice{Wrong}
41     \correctchoice{Right}
42     \correctchoice{Right}
43   \end{choices}
44 \end{questionmult}
45 }
46
47 \element{qqs}{
48 \begin{questionmult}{all for 2}\scoring{haut=2}
49   Perfect response scored 2 points, and give back one point for any error...
50   \begin{choices}
51     \correctchoice{Right}

```

[illegible]


```
112 \end{document}
```

Global scoring strategy

To use a strategy globally for a set of questions, one can define it in a LaTeX command, as in the following example:

```
\def\barQmult{haut=3,p=-1}

\begin{questionmult}\scoring{\barQmult}
[...]
```

Another possibility comes with the LaTeX commands **\scoringDefaultS** and **\scoringDefaultM**, to be used in the begin of the document (outside the command **\onecopy**), which allow to give default values for the scoring strategy of simple and multiple questions:

```
\scoringDefaultM{haut=3,p=-1}
```

In some cases, defining a global strategy can be interesting *depending of the number of proposed responses*. To do so, just input the value N. For example, to get a scale yielding 4 as the maximal score and such as the mean expected score of a student checking randomly the boxes is 1, one can use the scale $d=4, b=0, m=-(4-1)*2/N$ (which give the score -2 if every response is false, i.e. the wrong boxes have been checked and the right boxes are not). Operations allowed in those formulas are the four simple operations (+ - * /), cases operator (? :), parenthesis and all perl operators.

Other variables can also be used:

- N is the number of proposed responses, without counting the response eventually added by the option `completemulti`.
- NB is the number of correct responses to the question (without taking in account checked or non-checked boxes).
- NBC is the count of correct responses which have been checked.
- NM is the number of wrong responses to the question (without taking in account checked or non-checked boxes).
- NMC is the count of wrong responses which have been checked.
- IS is set to 1 if the question is simple and 0 if not.
- IMULT is set to 1 if the question is multiple and 0 if not.

From scoring strategy to students marks

Here is how students' marks are computed: for every student,

1. The scoring strategy is applied for each question in turn, to get the *questions scores*.
2. All questions (except indicative ones) scores are added to get the student *total score*.
3. If a positive *maximal mark* is given as a parameter (in the Project tab of the Edit → Preferences window), the total score of the student is divided by the *maximum total score* (which is the total score for a perfect copy), and multiplied by the *maximum mark*, to get the student's *mark*. This way, if the student answered perfectly to all questions, his mark will be the maximum mark. If you set the maximum mark to 100, the student's mark can be seen as a percentage of good answers.

Identification of the students

This stage is not mandatory. It deals with associating each copy with a student. The name of the student is not read in an automated fashion, but two reasonable possibilities are proposed:

1. It is possible to ask students to identify themselves on their copy with their student number, which will be written by checking one box per digit. A LaTeX command is designed to use this method on the copy (see the part Section 5.4.10). After the exam, copies will be identified automatically taking into account a list matching the students' numbers and their names.
2. With no input of the students' numbers, or in the case when the automated identification has not succeeded perfectly (for example when a student made a wrong input), the graphical interface allows an assisted manual association.

Let's first move to the Marking tab of the graphical interface.

List of the students

We must previously supply a list of students. This list can obviously be used for many multiple-choices tests. This list is a CSV file with optional comments lines at the beginning with prefix '#', as in the following example:

```
# STUDENTS / 1ST YEAR
surname:name:id:email
Bienvenüe:Alexis:001:paamc@passoire.fr
Boulx:Jojo:002:jojo.boulx@rien.xx
Noël:Père:003:pere.noel@pole-nord.xx
```

The first lines of the file which begin with the character '#' are comments. The first of the other lines contains (separated by the character ':') the column titles. Then, with one line per student, we write the corresponding information. There must be at least one column named `name` or `surname`.

Note

One can replace the separator ':' by a comma, a semicolon or a tabulation. However the same separator must be used everywhere in the file which contains the list of students. The used separator is detected by taking the character (out of the four possible characters) which appears most frequently in the first line which is not a comment. Any CSV file should be suitable.

The prepared list of students will then be selected with the button Set file in the Students identification section. We must also choose one of the columns as a unique key which will identify the students (generally, one chooses the column containing the student's number). Last, to prepare an automated association, we must choose the name of the relevant code used in the LaTeX command **AMCcode** (if used).

Association

Automated association

When we push the button Automatic in the part Students identification, matching of the codes given by the students begins. We can watch or improve the result later with a (partial) manual association.



Warning

To make an automated association, at least one command **AMCcode** is required (see Section 5.4.10) in the LaTeX source file, as well as a list of students with a column containing a reference (generally a number of student) which will be identical to the input given in the boxes produced by the command **AMCcode**.

Manual association

To open the window allowing recognition of the students' names, let's click on Manual button in the Students identification section. This window is made of an upper part which presents in sequence images of the names written by the students, a lower part containing a button for each student from the list we supplied, and a right part allowing to browse easily the copies to be identified. Let's click the button matching the name written in the upper part for each presented page (by default, only the copies not or badly identified are presented - this can be changed by checking the box associated). When every page is read, a blue background appears instead of the names, and we just need to click the Save button to end with association.

Exporting the scores list

At this stage, we can get the list of scores under various formats (currently CSV and OpenOffice), with the button Export. This export will be followed by the opening of the produced file by the appropriate software (if available).

Export to ODS (OpenOffice, LibreOffice)

In the exported file, the following colors are used:

grey is used for *non applicable*. This may be for example scores for absentees, or scores corresponding to a question that was not shown to the corresponding student.

yellow is used for questions that has not been answered by the student.

red is used for questions with an *invalid* answer: the student ticked several boxes in a simple question, or he ticked some boxes and the box *None of these answers are correct*.

purple used for indicative questions.

Annotation

When we push the button Annotate papers, copies annotation will begin: on every scan, the following annotations will be made (these are the default annotations, they can be configured):

- The boxes wrongly checked by the student will be circled in red;
- the non-checked boxes which should have been are checked in red;
- the checked boxes which had to be checked are checked in blue;
- for each question, obtained and maximal scores are indicated;
- the global score of the copy is indicated on the first page of the copy.

The text written on the first page of the copies can be configured (Edit → Preferences → Annotation → Header or Edit → Preferences → Project → Papers annotation → Header text). Substitutions will be made within the provided text (please have a look at Section 6.5.4 for some details on the meaning of those values):

%S is replaced by the student's *total score*.

%M is replaced by the *maximum total score*.

%s is replaced by the student's *mark*.

%m is replaced by the *maximum mark*.

%(ID) is replaced by the student's name.

%(COL) is replaced by the value of column *COL* in the students list for the current student.

This operation is made for each page, giving as a result PDF annotated papers. The name of the PDF file which will contain the corrected copy of a student is based on the template indicated in the field File name model. In that template, every substring as « *(col)* » is replaced by the contents of the column named *col* in the file containing the list of students (see section [List of the students](#)). If we let this field empty, a default value is built up based on the student's name and student number.

Alternative usages

Photocopied subject

As explained in [Printing and exam](#), it is not always possible to photocopy one answer sheet to give to several students. However, when using a separate answer sheet and when questions and answers are not to be shuffled, you can photocopy the subject, and print all the answer sheets separately. We detail here the proper way to follow.

- Use `separateanswersheet` package option (see [Package options](#)).
- Write the subject before calling **onecopy** command or outside **examcopy** environment.
- Use **\AMCformS** to output answer boxes in each answer sheet, inside **onecopy/examcopy**.

Here is a minimal exemple:

```
\documentclass[a4paper]{article}
\usepackage[separateanswersheet]{automultiplechoice}
\begin{document}

\noindent{\bf Subject}

\begin{question}{sum}
  How much are one plus one?
  \begin{choices}
    \wrongchoice{1}
    \correctchoice{2}
    \wrongchoice{3}
  \end{choices}
\end{question}

\begin{question}{k2}
  How high is the K2?
  \begin{choices}
    \wrongchoice{around 8000m}
    \correctchoice{around 8600m}
    \wrongchoice{around 9000m}
  \end{choices}
\end{question}

\AMCcleardoublepage

\onecopy{5}{

\AMCformBegin

{\large\bf Answer sheet:}
\hfill \namefield{\fbox{
  \begin{minipage}{.5\linewidth}
    Name:

    \vspace*{.5cm}\dotfill
    \vspace*{1mm}
  \end{minipage}
}}

\AMCformS

}

\end{document}
```

You will get from this LaTeX file one subject (sheet numbered 0) to print and photocopy to all students, and several answer sheets to print (one for each student).

Post correcting

Suppose you want to use a single generic answer sheet for all your exams. You simply print answer boxes on it (say 5 for each questions, and 40 questions), and give the students a subject that you wrote somewhere else. The point here is that the correct choices are not pointed out in the LaTeX file, so that AMC does not know about them. The solution is to give one answer sheet to the teacher to fill correct choices. Then, after scanning and AMC analysis, you simply have to tell AMC which is the teacher completed answer sheet.

To implement this idea, follow these rules:

- Use `postcorrect`, `insidebox` and `noshuffle` package options (see [Package options](#)).
- Only use `\wrongchoice` for all your choices (never `\correctchoice`).

Here is a minimal exemple:

```
\documentclass[a4paper]{article}
\usepackage{multicol}
\usepackage[insidebox,noshuffle,postcorrect]{automultiplechoice}

\begin{document}

\onecopy{5}{

\noindent
\begin{tabular}{|l|l|l|}
\hline
student number & class & subject\\
\hline
\hspace{-0.25cm}
& & \\
\AMCcode{StudentNum}{10}&
\AMCcode{class}{2}&
\AMCcode{subject}{3}
\\
\hline
\end{tabular}
\hfill\namefield{\fbox{
\begin{minipage}{.25\linewidth}
Name:

\hspace*{.5cm}\dotfill

\hspace*{.5cm}\dotfill
\hspace*{1mm}
\end{minipage}
}}\hfill

\hspace{.5cm}
\noindent\hrulefill

\begin{multicols}{2}\columnseprule=.4pt

\begin{question}{01}
\begin{choicescustom}
\wrongchoice{}%
\wrongchoice{}%
```

```

\wrongchoice{ }%
\wrongchoice{ }%
\wrongchoice{ }%
\end{choicescustom}
\end{question}

\begin{question}{02}
\begin{choicescustom}
\wrongchoice{ }%
\wrongchoice{ }%
\wrongchoice{ }%
\wrongchoice{ }%
\wrongchoice{ }%
\wrongchoice{ }%
\end{choicescustom}
\end{question}

% continue here to get as many questions as needed...

\end{multicols}

}

\end{document}

```

You can then process the LaTeX file in AMC, print the sheets, scan them after the exam, and start AMC automatic data capture (including the teacher sheet). When you click on Mark in the Marking tab, letting Update marking scale ticked, you will be prompted for the teacher answer sheet number. You can then continue as usual.

Note

You can also write the letters outside the boxes: replace the `insidebox` option with `outsidebox`, and write your questions in the following way:

```

\begin{question}{01}
\begin{choicescustom}
\wrongchoice{A }%
\wrongchoice{B }%
\wrongchoice{C }%
\wrongchoice{D }%
\wrongchoice{E }%
\end{choicescustom}
\end{question}

```

Nominative sheets

In some situations, it can be useful to prepare nominative sheets for all students, from a list of students. Let us see how this can be done.

- The students list has to be a CSV list. Suppose in the following that the file `students.csv`, in the project directory, is UTF8 encoded and that its content is like the following:

```

name,surname,id
Boulix,Jojo,001
Golin,André,002
Moniuszko,Stanisław,003

```

- The LaTeX source file has to load `csvsimple` package, with:
-

```
\usepackage{csvsimple}
```

- In the LaTeX source file, define the subject as a command that produce a single subject. This command will be called once for each student by **\csvreader** (suppose here that the questions has already be defined and included in a group named `general`):

```
\newcommand{\subject}{
  \onecopy{1}{

    \noindent{\bf AutoMultipleChoice \hfill TEST}

    \vspace*{.5cm}

    \begin{center}\em
      Pre-filled test.
    \end{center}

    \hfill \namefield{\fbox{
      \begin{minipage}{.5\linewidth}
        Name:

        \Large\bf \name{} \surname{}

        \vspace*{1mm}
      \end{minipage}
    }}

    \noindent\hrulefill

    \vspace{1ex}

    \shufflegroup{general}
    \insertgroup{general}

    \AMCassociation{\id}
  }
}

\csvreader[head to column names]{students.csv}{}{\subject}
```

The head to column names option for **\csvreader** defines commands **\name**, **\surname** and **\id** (named from the CSV headers), that can be used inside **\subject**. The **\AMCassociation** call tells AMC to associate the current sheet to student with id **\id**.

- After printing, scanning, data capture and marking, when associating copies with students, choose value *"pre-association"* for field "code-name for automatic association", and *"id"* for field "Primary key".

Commands manual

You may skip this part if you want to use only the graphical user interface (and that should usually be the case). However, every action carried out with the graphical interface can also be performed using the various commands whose syntax is described here.

auto-multiple-choice

auto-multiple-choice — Automated treatment of MCQs

Synopsis

```
auto-multiple-choice action [arguments...]
```

```
auto-multiple-choice [project]
```

Description

The **auto-multiple-choice** command transmits its arguments to the **AMC-*action*.pl** command.

The second form, which does not mention any action, calls the graphical interface **AMC-gui.pl**, possibly with the name of a project to be opened.

See also

Different AMC actions: AMC-prepare(1), AMC-imprime(1), AMC-analyse(1), AMC-note(1), AMC-association-auto(1), AMC-export(1), AMC-annotate(1), AMC-regroupe(1).

AMC-prepare

AMC-prepare — prepares working documents from LaTeX source file

Synopsis

```
auto-multiple-choice prepare --mode s --prefix project-dir mcq-source-file
```

```
auto-multiple-choice prepare --mode b --data project-data-dir mcq-source-file
```

Description

The **AMC-prepare.pl** command extracts working documents from the source file describing the multiple choice questionnaire. Information to be extracted depends on the `--mode` argument value. In any mode, the source filename must be given as an argument.

- with "`--mode s`", **AMC-prepare.pl** makes the subject file (answer sheet to be printed and given to the students), the correction (single corrected answer sheet) and the positions file (file containing information about the positions of the boxes on the pages). The following arguments can be used:

`--out-sujet subject.pdf` sets the subject file to build.

`--out-corrige corrected.pdf` sets the correction to build.

`--out-calage calage.xy` sets the positions file to build.

`--prefix directory` this directory is only necessary if one or more of the three previous options are not used: default values are then `directory/sujet.pdf`, `directory/corrige.pdf` and `directory/calage.xy`.

- with "`--mode b`", **AMC-prepare.pl** extracts scoring data from the source file. In this mode, the `--data` argument must be used (see below).

The following optional arguments can be used in any mode:

`--with latex-engine` gives the LaTeX engine (command) to be used. *latex-engine* can be **pdflatex** or **xelatex** for example.

`--filter filter` sets the filter name to transform the MCQ source file into a LaTeX file. Native filters are `latex` (no change at all) and `plain` (source is an AMC-TXT file).

--filtered-source *mcq-latex-file* gives the LaTeX file to make from the source file using specified filter. If omitted, a filename derived from *mcq-source-file* adding `_filtered.tex` is used.

--n-copies *n* sets the number of copies to produce, overriding the number given in the LaTeX source file (first argument of the `\onecopy` command).

--debug *file.log* gives a file to fill with debugging information.

--data *project-data-dir* sets the directory where the project's SQLite data files are.

AMC-meptex

AMC-meptex — gets the layout information from the working document to the layout database

Synopsis

```
auto-multiple-choice meptex --src calage.xy --data directory
```

Description

The **AMC-meptex.pl** command extracts the layout information (exact positions of the boxes, the marks, the name field on all the pages) from a working document *calage.xy* and fills the layout database (a SQLite file) in the data directory *directory*.

AMC-imprime

AMC-imprime — prints AMC multiple choice answer sheets to be distributed to the students

Synopsis

```
auto-multiple-choice imprime --sujet subject.pdf --fich-nums numbers-file.txt --data data-dir --methode method [where-to-print-arguments...]
```

Description

The **AMC-imprime.pl** command prints selected copies from a AMC multiple choice answer sheet.

What to print

The following arguments describe what to print:

--sujet *subject.pdf* sets the subject file (prepared by AMC-prepare(1)).

--fich-nums *numbers-file.txt* gives a file where the numbers of the copies to be printed are written (one number per line). If this argument is not given, all the copies will be printed.

--data *data-dir* gives the directory where data files are (see for example AMC-meptex(1)). The layout database in the data directory is used to know at which page of the subject file each copy begins and ends.

--split asks to print separate answer sheets separately.

Where to print

Several printing methods are currently defined:

- with "`--methode CUPS`", **AMC-imprime.pl** prints to a CUPS printer. One print job is sent for each copy, allowing for exemple to use stapling. Use the following options with this method:
 - `--imprimante printer` sets the CUPS printer name to print to.
 - `--options cups-options` gives CUPS options, in the `opt1=value1,opt2=value2,...` format.
- with "`--methode file`", **AMC-imprime.pl** outputs the answer sheets to files (one for each copy).
 - `--output filename` sets the filename for outputs. The `'%e'` sequence will be replaced by a 4-digits copy number. If `filename` does not contain `'%e'`, the string `'-%e.pdf'` will be added at its end.
- with "`--methode command`", **AMC-imprime.pl** will use a provided command for each copy.
 - `--print-command command` gives the command to be used for printing. The `command` string will be split at each space character (even when using quotes...). The sequence `'%f'` will be replaced by a PDF filename (containing the copy to print), and `'%e'` will be replaced by the copy number.

AMC-getimages

AMC-getimages — prepares scan images before sending them to AMC-analyse(1) for analysis

Synopsis

```
auto-multiple-choice getimages [--copy-to project-scans-dir] [--vector-density density] --list list-file [scan-files]
```

Description

The **AMC-getimages.pl** command prepares scan files before sending them to AMC-analyse(1):

- multipage image files are split to get one file per page.
- vector images (such as PDF, EPS) are converted to bitmap images.

The input images can be given either as arguments to AMC-getimages.pl, either as the name of a file which contains all the pathes of the scan files.

`--list list-file` gives the name of the file that optionnaly contains (one per line) the pathes of the scan files. This file will be cleared and filled with the pathes of the scan files after processing, so that the same path can be passed to **AMC-analyse.pl** as the `--list-fichiers` option value.

`--copy-to project-scans-dir` gives a directory where to copy all the scans files.

`--vector-density density` sets the density used to convert vector graphics scans to bitmap files. Defaults to 300.

AMC-analyse

AMC-analyse — automatic data capture from scans for AMC multiple choice exams.

Synopsis

```
auto-multiple-choice analyse --projet project-dir [--seuil-coche threshold] [--tol-marque tol] [--list-fichiers files-list.txt | scan-files]
```

Description

The **AMC-analyse.pl** command performs automatic data capture from scans of completed answer sheets of an AMC multiple choice questionnaire. Before calling **AMC-analyse.pl**, **AMC-prepare(1)** should have been called to make working documents (`--mode s`) and the layout should have been processed with **AMC-meptex(1)**.

--data data-dir gives the directory where data files are (see for example **AMC-meptex(1)**). Defaults to *project-dir/data*.

--cr project-cr-dir gives a directory where to create image reports of the data capture (zooms on the boxes in a zooms sub-directory, name filed extraction as a name-*.jpg file, and layout report as a page-*.jpg file).

--projet project-dir When using this option, if **--cr** is not used, *project-cr-dir* will be set to *project-dir/cr*.

Giving the names of the scan files to process can be done in two ways:

- these names can be given as arguments,
- these names can be written to a plain file (one filename per line), using the `--list-fichiers files-list.txt` to give the name of this file.

Some parameters for data capture may be set using the following options:

--prop proportion sets the proportion of each box that will be measured to decide if the box is ticked or not. Default value is 0.8.

--bw-threshold threshold This parameter is used when converting the grayscale scan file to black and white. To get more black pixels, use a greater value. To get more white pixels, use a smaller value. The threshold must be between 0 and 1. The default value is 0.6.

--ignore-red With this option, only red channel will be used from color scans. This way, all that is written in red will be ignored in the scan. This can be useful when the boxes are printed in red.

--tol-marque tol defines the tolerance when detecting the four marks in the scans corners. These marks are detected looking for black connected components which dimensions are *closed* to the target dimension *target* (exact dimensions of the marks if printing/scanning process where perfectly accurate). If *tol* is a real number, accepted dimensions are those between $(1-tol)*target$ and $(1+tol)*target$. If *tol* is "*tiny,tsup*" (where *tiny* and *tsup* are real numbers), accepted dimensions are those between $(1-tiny)*target$ and $(1+tsup)*target$. A standard value is 0.2.

--multiple This option is to be used when the scans are photocopies from some subjects (different students can have the same subject). In this case, *copy* numbers are allocated to students so that their answer sheets, with the same subject number, can be differentiated.

--pre-allocate copy_id This option can be used in conjunction with **--multiple**. The copy numbers used for the scans will start from *copy_id*, in the same order than the scans given as arguments.

--debug file.log gives a file to fill with debugging information.

AMC-note

AMC-note — computes marks after scans data capture for AMC multiple choice exams.

Synopsis

```
auto-multiple-choice note --data project-data-dir [--seuil threshold] [--grain granularity] [--arrondi rounding]
[--notemin min] [--notemax max] [--no-plafond | --plafond]
```

Description

The **AMC-note.pl** command computes marks for all students from the scoring strategy extracted from the LaTeX source file by AMC-prepare(1) and from the data capture reports made by AMC-analyse(1).

--data *project-data-dir* gives the directory where data files are (see for example AMC-meptex(1)).

--seuil *threshold* gives the black ratio threshold for deciding whether a box is ticked or not. When deciding whether a box is checked or not, **AMC-note.pl** compares the black ratio (number of black pixels over total number of pixels) to *threshold*. If the black ratio is greater than *threshold*, the box is declared to be checked. Standard values can be 0.15 in the standard layout, or 0.5 for separate answer sheet layout (in this last case, letters are drawn in the boxes, and the students are told to fill the boxes entirely).

--grain *granularity* --arrondi *rounding* ask marks to be rounded to a multiple of *granularity*. If *rounding* is 'i', rounding is done from below (as with floor(3)). If *rounding* is 'n', rounding is done to the nearest multiple of *granularity*. If *rounding* is 's', rounding is done from above (as with ceil(3)). For example, with options "**--grain 0.25 --arrondi s**", mark 6.285 is rounded to 6.5.

--notemin *min* with this option, all marks below *min* will be replaced by *min*.

--notemax *max* gives the mark to associate to a sheet where all answers are correct. If not used, marks are not scaled.

--plafond with this option, all marks above *max* will be replaced by *max*.

--debug *file.log* gives a file to fill with debugging information.

--postcorrect-student *s* --postcorrect-copy *c* requests port-correction from the completed answer sheet identified by student and copy numbers. In post-correction mode, correct answers are not extracted from the LaTeX source file, but taken from the answers given on this sheet.

AMC-association-auto

AMC-association-auto — automatic association between students and answer sheets for AMC multiple choice exams.

Synopsis

```
auto-multiple-choice association-auto --data project-data-dir --notes-id id --liste students-list.csv [--encodage-  
liste list-encoding] --liste-key key
```

Description

The **AMC-association-auto.pl** command associates students with their answer sheet (when there are no errors from students when coding their student number and no error during data capture). See Section 6.6 from english user documentation for details.

--data *project-data-dir* gives the directory where data files are (see for example AMC-prepare(1)).

--notes-id *id* gives the identification string of the code provided for student numbers (command **\AMCcode** in the LaTeX source file).

--pre-association if used, the **--notes-id** option is ignored and the automatic association is made from the pre-association data (command **\AMCassociation** in the LaTeX source file).

--liste *students-list.csv* gives the students list.

--encodage-liste *list-encoding* gives the students list file encoding (default is utf-8).

--liste-key *key* gives the column name where to find the student number in the students list.

--debug *file.log* gives a file to fill with debugging information.

AMC-association

AMC-association — manual association between students and answer sheets for AMC multiple choice exams.

Synopsis

```
auto-multiple-choice association-auto --data project-data-dir --list
auto-multiple-choice association-auto --data project-data-dir --set --student student-sheet-number [--copy
copy-number] --id student-id
```

Description

The **AMC-association.pl** command handles association data between students and their answer sheet.

--data *project-data-dir* gives the directory where data files are (see for example AMC-prepare(1)).

With option **--list**, all association data is output.

With option **--set**, a manual association is updated.

AMC-export

AMC-export — exports marks for AMC multiple choice exams.

Synopsis

```
auto-multiple-choice export --data project-data-dir --module module --fich-noms students-list.csv [--noms-
encodage list-encoding] --o output-file
```

Description

The **AMC-export.pl** command exports marks from an Auto Multiple Choice exam.

--data *project-data-dir* gives the directory where data files are (see for example AMC-prepare(1)).

--module *module* selects a module for export. See below for the modules included in the AMC standard distribution.

--fich-noms *students-list.csv* sets the students list file.

--noms-encodage *list-encoding* selects an encoding for file *students-list.csv* (default is utf-8).

--o *output-file* gives the output file name.

--option-out *options* gives an option for selected module, in the form *key=value* (see below for possible options for each module). To specify multiple options, use **--option** several times.

--sort *sort-type* sort the students names, depending on *sort-type*. If *sort-type* is *l*, use line number from students list file to sort. If *sort-type* is *m*, use mark (and name if marks are equal) to sort. If *sort-type* is *i*, use student number to sort. If *sort-type* is *n*, use name to sort (or line from students list if equal).

--useall *all* if *all* is 0 or empty, only students with a scanned answer sheet appears in the output. If *all* is 1, all students in the students list appears in the output.

Modules

OpenDocument

With "`--module ods`", an OpenDocument (for use by OpenOffice or LibreOffice for example) is produced. The following options are recognized:

nom gives a name for the exam, to be written on the beginning of the sheet.

code gives an short name for the exam, to be used as a tab name.

columns sets the list of columns that are to be added to each student. Default value is `student.key, student.name`.

stats with a true value, a table will be added with statistics for all questions.

statsindic with a true value, a table will be added with statistics for all indicative questions.

CSV

With "`--module CSV`", a CSV file is produced. The following options are recognized:

columns sets the list of columns that are to be added to each student. Default value is `student.copy, student.key, student.name`.

decimal sets the decimal point (default is a dot).

encodage sets the encoding to use for output (default is utf-8).

separateur sets the character used between columns (default is a comma).

entoure sets the character used to enclose strings (default is a double quote).

ticked if not empty, additional columns (with "`CHECKED :`" prefix) are included to give all the checked boxes on all the sheets. This can be used by an external program to retrieve all the checked boxes when marks are not sufficient. Use value `01` to get `0;0;1;1;0` if boxes 3 and 4 are checked, and value `AB` to get `CD` if boxes 3 and 4 are checked.

List

With "`--module List`", a PDF file is produced with a list of marks for all students. The following options are recognized:

pagesize The page size. Default value is `a4`.

nom gives the name of the exam, to be printed at the top of the page.

ncols The number of columns. Default value is `2`.

decimal sets the decimal point (default is a dot).

AMC-annotate

AMC-annotate — completed answer sheets annotation after marking for AMC multiple choice exams.

Synopsis

```
auto-multiple-choice annotate --projet project-dir --data project-data-dir --fich-assoc assoc.xml [annotation options...]
```

Description

The **AMC-annotate.pl** command annotates papers scans with scores for all questions, global score and mark, resulting in JPEG files (one for each page).

--projet *project-dir* sets the project directory.

--data *project-data-dir* gives the directory where data files are (see for example AMC-prepare(1)).

--cr *project-cr-dir* sets the project reports directory, where to find data capture reports prepared by AMC-analyse(1), and where to output annotated scans. Default value is *project-dir/cr*.

--fich-noms *students-list.csv* sets the students list file name.

--noms-encodage *encoding* selects an encoding for file *students-list.csv* (default is utf-8).

--qualite *qual* sets JPEG quality for output files. Default value is 65.

--taille-max *maxsize* sets the maximal size for output files (in the format *widthxheight*, in pixels). Default value is 1000x1500.

--pointsize-nl *lines* sets the size of texts requiring that with this size, the number of text lines that fits on the page height must be *lines*. Default value is 60.

--verdict *header* sets the text header to be printed on each first page of the students copies. This text may contain CR characters for multi-line headers. Some sequences are substituted in it:

%S is replaced by the student total score.

%M is replaced by the maximum total score.

%s is replaced by the student mark.

%m is replaced by the maximum mark.

%(ID) is replaced by the student's name.

%(COL) is replaced by the value of column *COL* in the students list for the curen student.

--verdict-question *perl-expression* sets the text to be printed next to all questions, through a perl expression to be evaluated (if you need a simple text, simply quote it with "). Some sequences are substituted in it before evaluation:

%S is replaced by the student score for this question.

%M is replaced by the maximum score for this question.

%s is the same as %S, but rounded to *nc* significant digits (see option **--ch-sign**).

%m is the same as %M, but rounded to *nc* significant digits (see option **--ch-sign**).

A standard value for this option is `"\"%s/%m\""`. You can also use tests (with the perl syntax (`TEST ?IF-YES : IF-NO`)) to write *Valid* if the score is the maximal score, or *Failed* if not, using

```
--verdict-question "(%S==%M ? \"Valid\" : \"Failed\")"
```

--ch-sign *nc* Sets the number *nc* of significant digits to use when printing global mark.

--symbols *symbols-definition* defines how boxes will be annotated. *symbols-definition* is a comma separated list of *H-A:shape:colour* constructs, where *H* is 0 or 1 depending on wether the box has to be checked or not, *A* is 0 or 1 depending on wether the box is actually checked or not, *shape* is one of *none*, *circle*, *mark*, *box* and *colour* is a colour (name or #RGB, see ImageMagick or GraphicsMagick documentation for details). Default value is `"0-0:none,0-1:circle:red,1-0:mark:red,1-1:mark:blue"`.

--position *pos* sets the position of individual questions scores. *pos* may be 'none' (do dot write individual scores), 'margin' (write scores in the margin) or 'case' (write scores at the left of the left-most box for the question).

--ecart *dx* sets the horizontal space to let between individual question score and the leftmost box when **--position** *case* is used.

--ecart-marge *x* sets the x position for individual questions scores when written in the margin (in *em* units).

--line-width *lw* sets the line width for symbols drawn on the scans (in pixels). Default value is 2.

--indicatives *bool* say if indicative questions are to be annotated (use 1 for a true value; default is false).

--debug *file.log* gives a file to fill with debugging information.

AMC-regroupe

AMC-regroupe — merge JPEG annotated AMC multiple choice answer sheets to get a PDF file per student.

Synopsis

```
auto-multiple-choice regroupe --projet project-dir --sujet subject.pdf --modele file-name-model --fich-noms
students-list.csv [--noms-encodage encoding] [--compose]
```

Description

The command **AMC-regroupe.pl** merges annotated AMC multiple choice answer sheets (there is one per page) to get a single PDF file per student.

--data *data-dir* gives the directory where data files are (see for example AMC-meptex(1)).

--projet *project-dir* Gives the project directory.

--sujet *subject.pdf* gives the subject file (as prepared by AMC-prepare(1)). This helps in getting the page size for the subject to make annotated PDF fit this size.

--fich-noms *students-list.csv* sets the students list file name.

--noms-encodage *encoding* selects an encoding for file *students-list.csv* (default is utf-8).

--modele *file-name-model* sets a file name model for annotated PDFs. In this model, some sequences will be substituted:

- (N)** is replaced by the student's name.
- (ID)** is replaced by the student number.
- (COL)** is replaced by the value of column *COL* in the students list for the current student.

The default value is '*(N) - (ID) .pdf*'.

--compose When using this option, pages for which there is no scan will be replaced by the corresponding page from the corrected answer sheet. This option is for example useful when using the separate answer sheet layout: in this case, the annotated answer sheet will not be readable without the pages with the questions.

When using the **--compose** option, one must also use the **--tex-src** (and optionally **--with**, **--filter** and **--filtered-source**) option.

--tex-src *mcq-source-file* gives the source file for the subject.

--with *latex-engine* gives the LaTeX engine (command) to be used to process the (optionally filtered) source file. *latex-engine* can be **pdflatex** or **xelatex** for example.

--filter *filter* sets the filter name to transform the MCQ source file into a LaTeX file (see AMC-prepare(1)).

--filtered-source *mcq-latex-file* gives the LaTeX file to make from the source file using specified filter (see AMC-prepare(1)).

--debug *file.log* gives a file to fill with debugging information.

AMC-mailing

AMC-mailing — mail PDF annotated completed answer sheets to students

Synopsis

```
auto-multiple-choice mailing [--xmlargs args.xml] --project project-dir --students-list students-list.csv [--list-encoding encoding] --email-column col --sender sender-email --subject subject --text email-body [--debug file.log] [transport arguments...]
```

Description

The command **AMC-mailing.pl** sends by email the PDF annotated completed answer sheets to the students. Email addresses are taken from the students list file.

--project *project-dir* Gives the project directory.

--students-list *students-list.csv* sets the students list file name.

--list-encoding *encoding* selects an encoding for file *students-list.csv* (default is utf-8).

--email-column *col* sets the name of the column containing the email addresses of the students in the students list file.

--sender *sender-email* sets the sender email.

--subject *subject* sets the subject of the emails to send.

--text *email-body* sets the body text of the emails to send.

--attach *filename* adds file *filename* as an attachment to all emails sent. Use this option several times to attach multiple files.

--transport *transport* sets the transport to use. *transport* may be `sendmail` or `SMTP`.

--debug *file.log* gives a file to fill with debugging information.

Transport arguments

Depending on the value used for argument *transport*, the following arguments can be added:

Transport sendmail

--sendmail-path *path* sets the path to the **sendmail** command (defaults to `/usr/sbin/sendmail`).

Transport SMTP

--smtp-host *host* sets the SMTP hostname (defaults to `smtp`).

--smtp-port *port* sets the port to use (default value is 25).

Arguments in a XML file

To avoid encoding problems, one can place some arguments in a XML file and use the `--xmlargs file.xml` argument as the first command argument. Here is an exemple of such a file:

```
<?xml version="1.0" encoding="UTF-8"?>
<arguments>
  <arg>--sender</arg><arg>Bienvenüe &lt;paamc@passoire.fr&gt;</arg>
  <arg>--text</arg><arg>Voilà votre copie corrigée</arg>
  <arg>--subject</arg><arg>QCM</arg>
</arguments>
```