Sending
Answers to Exercises
(or Proofs of Theorems)
to appendices

(Version 1.2.1, last modified 2019/07/24)

Didier Rémy

July 24, 2019

Abstract

This package provides macros for writing exercises and answers to exercises. Then, the same document can be run in several modes, for instance, the answers may be hidden, inserted on place, or just collected in an auxilliary file to be inserted later in an appendix for instance. (Then hyperref can be used to make questions and answers point to one another.)

The package can also be used to send proofs of theorems to appendices.

The package also comes with a version for HEVEA.

1 The exercise and answer environment

The `exercise` environment is used to separate and number exercises. The environment contains an optional parameter meant to be a short description of the exercise that will then appear as the title of the exercise, when refered to from another position.

The `answer` environment is normally used within the `exercice` environment, in which case the exercise number is captured and used to refer to the exercise from another position. There may be several answers within a single exercise environment. Answers are numbered internally (*ie.* independently of any question and subsection structure that the author could introduce) and its number refers to the n-th answer envrionment from the beginning of the exercise.

1.1 Collecting answers into an answer-file (default mode)

The default behavior of the `answer` environment is to copy the contend of the answer (the region of text between `\begin{answer}` and `\end{answer}` verbatim to the answer

file \ExerciseAnswerFile.ans (by default \ExerciseAnswerFile is equal to \jobname)

The answer environment can also be used outside of the exercise environment, in which case the text is inserted verbatim in the answer file, but without any title and any other side effect. For instance, this may be used to insert in the answer files some headers or notes that are global to several exercises.

The fragile command \fragileanswer is an alternative to the use of the use of the answer environment. However, since it evaluates its arguments, commands and such must be quoted. The advantage is that it can be used inside other macros.

The command \inputanswers normally insert all answers to exercises. It takes the title of the answer section as a mandatory argument. The base file name of the answer file, may be given as an optional argument; this allows to load answers generated from another file.

1.2 Customizing answer placements

Customization of answer placement uses the keyval package. The command \setanswer can be used with the following keys:

noanchor

> This will remove hyperref anchors to exercises (which are set by default).

anchor

> This will attached hyperref anchors to exercises (if previously removed).

hide

> Answers will be hidden.

show

> Answers will be shown.

later

> If shown, answers will appear later.

inline

> If shown, answers will appear immediately.

file=*filename*

> This will use filename as the base name for the answer file. By default \jobname is used. The suffix .ans will be appended to the basename. This command key can only be set in the preamble.

The command `\inputanswers` can be used to insert answers that have been postponed. The command takes one argument which is used as the title of the answer section before answers are inserted. The command also takes an optional argument which is the name of the answer file. By default the filename if the same one as the one in which answers were collected. If answers are ignore (as set by the `hide` key) or immediate (as set by the `inline` key) at the time `\inputanswers` is executed, then the command is ignored (hence, the answer section is not created).

These keys can be set globally with the command `\setanswers`. The can also be modified globally anywhere or locally in the scope of grouping parens. Furthermore, they can be set locally to an exercise by appearing in an optional argument to the exercise environment.

1.3 Package options

The keys `noanchors`, `hide`, and `now` are also recognized as package options. Any other package option is interpreted as the argument to the key `file`.

1.4 Example

Exercise 1 *What would be the effect of the following declaration?*

```
\let \oldsection \section
\renewcommand {\section}[1]
  {\oldsection {#1}\fragileanswer
     {\string \subsection* {Section \thesection\space (#1)}}}
```
Answer

(Of course the code need improvement to take the optional arguments sections into account) □

Exercise 2 *Note that this exercise has no title.*
1. what will you see at this place in the main body of the document?   Answer
2. Where will I see the answer to the previous question and to this one.   Answer □

Exercise 3 (Hiding answers) *What would happen if I type* `\setanswer{noanchor}` *in the body of this exercise?*   Answer
And what will happen if I now type `\setanswer{hide}` *in the body of this exercise?*
Answer □

Exercise 4 (Input Answers) *What will happen if the command* `\inputanswers` *is never included?*   Answer
Note that the file `\jobname.ans` *is only created if non empty. Thus, if you load the package but do not create any exercise it is not created. Then, you will simply get a warning if you attempt to load answers of exercises.*   □

1.5   Customizing the appearance of answers

This can be done by redefining the following commands:

`\AnswerSection`

> The name of the command to print the title of answer section (the argument of `\inputanswers`.

`\ExerciseName`

> The identifier used to name exercises (often need to be changed according to languages). Default value is `Exercise`, except if `\languagename` is `frenchb` where the value is `Exercice`.

`\AnswerName`

> The identifier used to point to answers. Default value is `Answer`, except if `\languagename` is `frenchb` where the value is `Réponse`.

`\AnswerNew`

> Typeset the title of an answer to a new exercise. It receives 2 arguments: the exercise number and the answer number.

`\AnswerContinued`

> Idem, but when the answer is not the first one of the exercise.

`\AnswerAnchorPage`

> Receives two arguments: the exercise and the answer numbers. Can be used in the definition of the two previous commands to find the page of the anchor of the given answer within the given exercise.

`AnswerNow`

> This environment can be redefined to change the typesetting immediate answers (when answers are immediate, Answer will act as AnswerNow).

2   Creating new answer environments

In addition to customization of the answer environment, the user can defined its own answer environment. For instance, an environment `myanswer` similar to the default `answer` of this package can be defined with the following definition:

```
\newanswer {myanswer}{Answer}
  {\em \AnswerNiceLink}
  {}
  {\AnswerNow}
  {\endAnswerNow}
  {{\thepage}}
  [1]
  {\ifnum \theQuestion=1\relax
      \subsection*
         {\ExerciseLink {Exercise \theExercise, page #1}}%
   \else
      \subsubsection*
         {\ExerciseLink {Exercice \theExercise\space (continued)}}%
   \fi}
  {}
```

The command `\newanwer` takes at least 8 arguments, with two possible optional arguments after the fourth one.

1. The first argument is the name of the answer environment being defined.

2. The second argument is the title of the answer environment to appear in the document.

3, 4. This are will be executed respectively before and after processing the answer. Typically, they are used to insert a pointer to the exercise in the appendix.

5, 6. These define the environment that should be used in case the answer is inserted locally, *ie.* if `\AnswerLaterfalse` is set. Two optional arguments may be used between mandatory arguments 4 and 5 and will act as in the `\newenvironment` command.

7. This is used to pass information between the place where the answer environment occurs and the time when the answer will be reread again. This argument is passed to the `\fragileanswer`, *ie.* it is evaluated in the local context but inserted in the answer file.

8, 9. The remaining arguments are used to define the behavior of the answer when reread from the answer file. They are passed to the command `\newenvironment{ans-name}` (where `name` is the name of the answer-environment being defined) hence the two optional arguments in case this command used arguments (possibly with default values).

Additionally, the so-created `myanswer` environment that is immediately passed to `\setanswer` for local customization. This is evaluated first, so that it act as if it `\setanswer` were executed outside of the exercise environement, except that the setting if local to the environment.

More precisely, at every of the follow code in the source file:

```
\begin{myanswer}[<args>]
<Any text>
\end{myanswer}
```

The following steps are executed in order (#k refers to the k's argument of the `myanswer` environment definition):

1. As at the beginning of any environment, a new group (scope) is opened

2. The optional argument `<args>` of the environment is passed to `\setanswer`. Unless the answer is hidden or immediate, the following steps continues as folllow (otherwise, the environment is ignore or inlined according to arguments `#4` and `#5`).

3. `\begin{ans-myanswer}` is inserted in the answer file.

4. The argument `#6` is inserted into the answer file.

5. The argument `#2` is evaluated.

6. The body of the answer `<Any text>` is inserted verbatim into the answer file.

7. The argument `#3` is evaluated.

8. `\end{ans-myanswer}` is inserted in the answer file.

9. The group (scope) of the environment is closed.

In addition, two commands are available for manipulating anchors: `\AnswerLink` can be used in the text to point to the corresponding answer in the answer section and, conversely, `\ExerciseLink` can be used to point to the exercise from the answer section. Both commands take one argument (the text where to anchor the link). A more elaborated version `\AnswerNiceLink` provides a default typesetting of the answer link. It takes an optional parameter —the text of the anchor— equal to `\AnswerName` by default.

The environment also adjusts the counters `\Exercise` and `\Question` so that their values while reading the answer file is identical to the value they had when the corresponding answer was processed. Other counters (such as pages, sections, etc.) must be passed explicitly using the fourth argument, as described above.

These steps only applies is the answer appears within an exercise, and if answers are not hidden. Otherwise, the environment behaves as in the default case.

3   Sending proofs to an appendix

This package can also be used to typeset proofs locally right after the claims, and have them appear in an appendix.

For instance this can be realized with the following declaration:

```
\def \capitalize #1{\uppercase {#1}}
\newanswer {proof}{Proof}
  {\def \theExercise {\ThmName}\def \theQuestion {\ThmLabel}%
   \if@answer@later
       \if@answer@anchor \AnswerLink{\hbox {(See proof in the Appendix)}}\fi%
   \else \def \AnswerName {Proof}\fi
   \pdf@hypertarget {Thm-\ThmName.\ThmLabel}{}}
  {}
  {\trivlist \item [\hskip \labelsep {\underline {Proof}:}]]}
  {\penalty 100\hfill\rule{2mm}{2mm}\endtrivlist\@doendpe}
  {{\ThmName}{\ThmLabel}}
  [2]
  {\subsection*
       {\pdf@hypertarget {Ans-#1.#2}{Proof} of
        \pdf@hyperlink {Thm-#1.#2}{\capitalize #1 #2}}%
   \nobreak
   }
  {}
```

Note the use of `\ThmName` and `\ThmLabel`, which refer to the name and counters of the last theorem-like environment. They can be retrieved automatically will the following little hack.

```
\def \@thm #1#2{%
  \refstepcounter {#1}
  \xdef \ThmName {\@currenvir}%
  \xdef \ThmLabel {\@currentlabel}%
  \@ifnextchar [{\@ythm {#1}{#2}}{\@xthm{#1}{#2}}%
}
```

Then, proofs will be recalled by

```
\inputanswers {Proof of main results}
```

Warning! Although the package can be used to pospone several kinds of material to an appendix using a temporary file, it uses a single file so far. So combining solutions to exercises and proofs to appendices will not work together (material would be interlaced). Sometimes interlacing may be desired, but usually not. In the future, the command should offer the possibility to send different kind of material to different files.

When, `\inputanswers` is called, all pending answers are inserted. If other (exercises and) answers occur after `\inputanswers`, the file will be reopened as an empty file and new answers will be

4   Implementation notes

The implementation is quite simple and short. The only trick is to undo the active cat-codes that might have been done by the `inputencoding` package before copying answers to `\ExerciseAnswerFile.ans`.

Beware!

However, I found out, that to make it work you also need to tell latex to recognize 8-bit characters, i.e. to treat "é" as such, and not as "^^e9". On my redhat 6.1 linux machine, it is sufficient to set the environment variable `LC_ALL=en_US`. I do not know any otherway of making latex print 8-bit characters. Of course, printing "^^e9" instead of "é" is quite problematic with with verbatim code, where the "^^" will be taken verbatim and not as an encoding of an 8-bit character. The solution would them be to pacth the verbatim mode to keep the meaning of "^^" as encoding of 8-bit charaters.

Another, probably simpler, solution would then be to post-process the answer file to recode 7-bit charaters into 8-bit characters. However, one cannot reread the reshly created answer file. Instead two different names for the output and input answer files should be used.

5   HeVeA compatibility

The package also comes with some HEVEA version. HEVEA asumes that the file has first been processed by tex so that `\jobname.ans` already exists.

However, HEVEA does not currently interpret package options. So, you must set the corresponding flags by hand, instead, to ensure HEVEA compatibility.

to change the name of `\jobname` does not currently work in HEVEA, so you will have to do the appropriate renaming by hand (or in a Makefile).

The HEVEA version is designed to be combined with hacha: in addition to pointing to answers, the answers are also included immediately, but in a separate cutting unit, so that hacha will put them in an auxilliary file.

6   Answers to exercises

Answers to exercises are intentionlly very brief.

Exercise 1, page 3

See the section information inserted above and below in the answer section.

Exercise 2, page 3

The question above, but you will not see this.

Exercise 2 continued, page 3

At the place where you insert `\inputanswers`

Exercise 3, page 3

The answer would be shown as usual in the list of answers, but there won't be an anchor from the exercise to its solution. Note that this would also affect following answers of the same exercise.

Exercise 3 continued, page 3

This answer would be invisible, as if removed from the file.

Exercise 4, page 3

The answer of exercises will just not be shown and (anchors to answers will be dandling pointers)

7   We continue, and may include more exercises

Exercise 5 (This is an additional optional exercise) *Where the answer to this exercise will go?*   Answer
How can we include answers of these additional exercises?   Answer □

8   Answers to additional exercises

Exercise 5, page 9

In a new file `\jobname1.ans`.

Exercise 5 continued, page 9

By just repeating the `\inputanswers` command. Further answers will be stored in file `\jobname2.ans`, etc. but you should keep using the same `\inputanswers` command.