

The `exercise`* package

Mark Wooding

25 August 2003

Contents

1	User guide	1	2.2	Checking for counter changes	5
1.1	Exercises and answers	1	2.3	The exercise environment	5
1.2	The answers file	2	2.4	Answers	6
1.3	Style tweaks	2	2.5	Lists inside paragraphs . .	7
1.4	Lists in paragraphs	3	2.6	Errors	8
1.5	And finally	3			
2	Implementation	4	A	The GNU General Public License	8
2.1	Initialization	4			

1 User guide

The `exercise` package allows you to typeset exercises and keep the answers together with the questions in your source (so they don't get lost) but typeset them all together at the end of your document.

1.1 Exercises and answers

`exercise` Exercises are typeset in an `exercise` environment. This takes no arguments. There's a counter for exercises, named `exercise`, and you can cross-reference exercises in the usual way.

`\answer` If you want to include an answer for your exercise, just say `\answer` followed by your answer. The rest of the text up to the end of the `exercise` environment is considered to be the answer, and is stored away until asked for. This material can contain anything you like. It *isn't* a moving argument.

`\skipanswer` The `\skipanswer` command is similar, but just skips over the answer rather than writing it to the output file. It does nothing at all with counters – it's as if there was no answer given at all.

It's OK to use commands like `\answer` and `\skipanswer` in your own macros as long as they're the *last* token. You can therefore say something like

```
\newcommand{\evenanswer}{%
```

*The `exercise` package is currently at version 1.1, dated 25 August 2003.

Exercise 1 Show that if $F: \{0, 1\}^k \times \{0, 1\}^* \rightarrow \{0, 1\}^L$ is a $(t, q + 1, \epsilon)$ -secure PRF, then F is also a $(t, q, \epsilon + 2^{-L})$ -secure MAC.

```

\begin{exercise}
Show that if  $F: \{0, 1\}^k \times \{0, 1\}^* \rightarrow \{0, 1\}^L$ 
is a  $(t, q + 1, \epsilon)$ -secure PRF, then  $F$  is also a
 $(t, q, \epsilon + 2^{-L})$ -secure MAC.
\answer
Let  $A$  be an adversary attacking  $F$  as a MAC. Consider the
adversary  $B$  \dots
\end{exercise}

```

```

\ifthenelse%
{\isodd{\value{exercise}}}%
{\answer}%
{\skipanswer}%
}

```

to get just the answers to the odd-numbered problems. (If you don't like `ifthen` then you'll need to play with `\expandafter` for a bit.)

1.2 The answers file

`\answrite` Answers are accumulated into a file, to be read later. Additional material can be added to the file using the `\answrite` macro, which just writes its argument. Note that this is a *moving* argument, so fragile commands need `\protecting`.

`\exctrcheck` It's common to divide up the answers by section. You can tell the package to check a collection of counters and perform actions if they've changed since last time, giving you a chance to write the correct decorations to the answers file. This is done by saying `\exctrcheck{<counter>}{<action>}`. Then, each `\answer` command checks to see if `<counter>` has changed since last time, and if it has, it does `<action>`. For example,

```

\exctrcheck{section}
{\answrite{\protect\subsection*{Section \thesection}}}

```

starts a new (unnumbered) subsection in the answers for each section in the main document.

1.3 Style tweaks

The exercise environment is very simple, and can be easily rewritten to fit in with your style preferences. If you like exercises to look like theorems, the easiest thing to do is say something like

```

\newtheorem{doexercise}[exercise]{\exercisename}
\renewenvironment{exercise}{\exfix\doexercise}{\enddoexercise}

```

This makes the environment use the existing exercise counter. If you don't want that, say

```
\newtheorem{doexercise}[othercounter]{\exercisename}
\renewenvironment{exercise}{\exfix\doexercise}{\enddoexercise}
\let\theexercise\theothercounter
```

and all will be well.

`doanswer` Answers are typeset in a `doanswer` environment, which is given one argument: the exercise number (as set by `\theexercise`). This can be modified to do whatever you like.

`\exfix` The `\exfix` is a convenient hook which is run both in the `exercise` and `doanswer` environments by default. The current implementation just skips a level of `enumerate` depth, which usually means that `enumerate` lists will go (a), (b), (c), ... rather than 1, 2, 3, ...

1.4 Lists in paragraphs

`parenum` Answers to subparts tend to be compressed together into a single paragraph. It's nice, when you do this, not to have to worry about losing your numbering of subparts. The `parenum` environment provides an enumerated list in a paragraph. So, for example, you can say something like this.

Example of `parenum`

Exercise 2 A PRG $g: \{0, 1\}^k \rightarrow \{0, 1\}^L$ is *trivial* if $k \geq L$.

- (a) Show that trivial PRGs exist.
 - (b) Show that a non-trivial (t, ε) -secure PRG is a $(t, \varepsilon + 2^{k-L})$ -secure one-way function.
-

```
\begin{exercise}
A PRG  $g$  colon  $\{0, 1\}^k$  to  $\{0, 1\}^L$  is \emph{trivial} if
 $k \geq L$ .
\begin{enumerate}
\item Show that trivial PRGs exist.
\item Show that a non-trivial  $(t, \varepsilon)$ -secure PRG is a
 $(t, \varepsilon + 2^{k-L})$ -secure one-way function.
\end{enumerate}
\answer
\begin{parenum}
\item The identity function is a trivial PRG: the real and random
games are identically distributed.
\item Let  $\mathcal{A}$  be an adversary attempting to invert  $g$ : then we
can construct a distinguisher  $\mathcal{B}$  as follows \ldots
\end{parenum}
\end{exercise}
```

1.5 And finally

`\answers` In order to extract your answers, say `\answers`.

Answers

1 Let A be an adversary attacking F as a MAC. Consider the adversary $B \dots$

2 (a) The identity function is a trivial PRG: the real and random games are identically distributed. (b) Let A be an adversary attempting to invert g : then we can construct a distinguisher B as follows \dots

```
\section*{Answers}
\answers
```

The `\answers` command has an optional argument, which is the file to read in. This allows you to make ‘answer booklet’ documents, by saying something like

```
\answers[otherdoc]
```

If you don’t give a file extension, then `.ans` is appended automatically.

2 Implementation

```
1 (*package)
```

2.1 Initialization

The `within` option is handled by the `mdwkey` package.

```
2 \RequirePackage{mdwkey}
```

`\ex@within` When the `within` option is seen, we set a command `\ex@within` to the correct code, to be executed later when we’ve made our minds up.

```
3 \let\ex@within\relax
4 \mkdef{exercise:opts}{within}{%
5   \def\ex@within{%
6     \@addtoreset{exercise}{#1}%
7     \toks@\expandafter{\csname the#1\expandafter\endcsname}%
8       \expandafter.\theexercise}%
9   \edef\theexercise{\the\toks@}%
10  }%
11 }
```

`\ex@opts` The `\ex@opts` macro just runs the `mdwkey` kit to parse an option string.

```
12 \def\ex@opts{\mkparse{exercise:opts}}
```

Now do the options thing.

```
13 \DeclareOption*{\expandafter\ex@opts\expandafter{\CurrentOption}}
14 \ProcessOptions*
```

Set up the `exercise` counter, and number it within some other sort of counter as appropriate.

```
15 \newcounter{exercise}\ex@within
```

We also need the `sverb` package in order to do the delaying of the answers.

```
16 \RequirePackage{sverb}
```

2.2 Checking for counter changes

`\ex@ctrcheck` The counter checking state is stored here. It's initially empty.

```
17 \def\ex@ctrcheck{}
```

`\exctrcheck` Adding a counter to the check list is relatively easy. We expand the current list into a token register, add the new material to the end, and put the list back in our macro using `\edef`. The 'last' value of the counter is set to `\relax`, to force out the change on the next `\answer`.

```
18 \def\exctrcheck#1#2{%
19   \toks@\expandafter{\ex@ctrcheck\ex@ctrdo{#1}{#2}}%
20   \edef\ex@ctrcheck{\the\toks@}%
21   \global\expandafter\let\csname ex@ctrlast@#1\endcsname\relax%
22 }
```

`\ex@ctrdo` Here we actually check to see whether a counter has changed and execute the appropriate code.

```
23 \def\ex@ctrdo#1#2{%
24   \edef\@tempa{\csname the#1\endcsname}%
25   \expandafter\ifx\csname ex@ctrlast@#1\endcsname\@tempa\else%
26     #2%
27   \global\expandafter\let\csname ex@ctrlast@#1\endcsname\@tempa%
28   \fi%
29 }
```

2.3 The exercise environment

`\exercisename` We store the string to print for each exercise in `\exercisename` as a half-hearted attempt at internationalization.

```
30 \providecommand\exercisename{Exercise}
```

`\exfix` This is a dumping ground for style fixing common to both exercises and answers. Here, we bump on the `enum depth` counter, so that it skips labelling with digits.

```
31 \def\exfix{\advance\@enumdepth\@ne}
```

`exercise` This is pretty simple. The environment is list-based, with the number set in bold in a label.

```
32 \def\exercise{%
33   \refstepcounter{exercise}%
34   \exfix%
35   \trivlist\advance\itemindent\labelsep%
36   \item[\textbf{\exercisename\ \theexercise}]%
37 }
38 \let\endexercise\endtrivlist
```

2.4 Answers

We need a file in which to store our answers.

```
39 \newwrite\ex@ansfile
```

`\ex@ansfilename` In case anyone has a better idea for a filename than our default, we provide a hook.

```
40 \def\ex@ansfilename{\jobname.ans}
```

We open the file at the end of the preamble, to give the user a chance to say `\nofiles`, or change `\ex@ansfilename`.

```
41 \AtBeginDocument{%
42   \if@filesw%
43     \immediate\openout\ex@ansfile=\ex@ansfilename\relax%
44     \answrite\relax%
45   \fi%
46 }
```

`\answrite` This writes stuff to the answers file. We make sure that it's appropriately protected, so that you can insert section headings and so on.

```
47 \def\answrite#1{%
48   \if@filesw%
49     \begingroup%
50     \let\protect\@unexpandable@protect%
51     \immediate\write\ex@ansfile{#1}%
52     \endgroup%
53   \fi%
54 }
```

`\answer` The `\answer` macro needs to read until the end of the enclosing exercise environment (or whatever).

```
55 \def\answer{\sv@readenv\ex@answer}
```

Now for the main guts.

```
56 \def\ex@answer#1#2{%
57   \begingroup%
58   \@bsphack%
```

First of all, check to see whether any counters have changed.

```
59   \ex@ctrcheck%
```

Start a `doanswer` environment in the file.

```
60   \answrite{\noexpand\begin{doanswer}{\theexercise}}%
```

Set catcodes to be strange, and read lines one-at-a-time, writing them to the file. When finished, continue at `\ex@endanswer`.

```
61   \let\do\@makeother\dospecials%
62   \sv@safespc%
63   \sv@read{#1}\answrite{\ex@endanswer#2}%
64 }
```

When that's done, we wind up here.

```
65 \def\ex@endanswer{%
66   \esphack%
67   \answrite{\noexpand\end{doanswer}}}%
68   \endgroup%
69 }
```

`\skipanswer` This is much simpler.

```
70 \let\skipanswer\ignore
```

`\answers` The `\answers` macro closes the file, makes sure that future exercises with answers cause an error, and reads in the file.

```
71 \def\answers{%
72   \@ifnextchar[{\answers@i\input}{\answers@i\input[\ex@ansfilename]}%
73 }
74 \def\answers@i#1[#2]{%
75   \immediate\closeout\ex@ansfile%
76   \global\let\answrite\exerr@toolate%
77   \ex@withext{#1}{#2}{ans}%
78 }
79 \def\q@delim{\q@delim}
80 \def\ex@withext#1#2#3{%
81   \edef\next@##1{\noexpand\ex@ext@i{##1}{#2}{#3}#2.\noexpand\q@delim}%
82   \next@{#1}%
83 }
84 \def\ex@ext@i#1#2#3#4.#5\q@delim{%
85   \ifx\q@delim#5\q@delim\def\next@{#1{#2.#3}}%
86   \else\def\next@{#1{#2}}\fi%
87   \next@%
88 }
```

`doanswer` A very simple environment. We set the exercise number in bold and then just write the text.

```
89 \def\doanswer#1{%
90   \exfix%
91   \trivlist\advance\itemindent\labelsep%
92   \item[\textbf{#1}]%
93 }
94 \let\endoanswer\endtrivlist
```

2.5 Lists inside paragraphs

`parlist` The `parlist` environment is a trimmed-down version of a normal list. We todge the `\item` command, make `\list` and `\trivlist` make errors, and do various normal list things.

```
95 \def\parlist#1#2{%
96   \let\trivlist\exerr@parlist%
97   \def\itemlabel{#1}%
98   \let\makelabel\relax%
99   \@nmbulistfalse%
100  #2%
101  \let\item\pl@item%
```

```

102 \ignorespaces%
103 }

\pl@item This is the implementation of \item within a parlist. The main interesting point
is the game with boxes, which has the objective of extracting the text of the item,
together with any style changes set by \makelabel, but without any stupid bits
of glue, or \llap or anything like that.

104 \def\pl@item{\@ifnextchar[\pl@item@i{\pl@item@i[\@itemlabel]}}
105 \def\pl@item@i[#1]{%
106 \if@mbbrlist\refstepcounter{\@listctr}\fi%
107 \setbox\z@\hbox{\makelabel{\global\setbox\@ne\hbox{#1}}}%
108 \ifvmode\leavevmode\else\unskip\hskip1em\fi\box\@ne~\ignorespaces%
109 }

\useparlist We just set the list environment to use parlist.

110 \def\useparlist{\let\list\parlist\let\endlist\relax}

parenum Very simple, this. Note that we don't run \endenumerate, because that's \let to
\endlist.

111 \def\parenum{\useparlist\enumerate}
112 \let\endparenum\endparlist

```

2.6 Errors

```

113 \def\exerr@toolate{%
114 \PackageError{exercise}{Too late now for \string\answrite}{%
115 You can't write answers after you've read the file in. I've^^J%
116 ignored the text you attempted to write. This is why answers^^J%
117 go at the end of a book!%
118 }%
119 }
120 \def\exerr@parlist{%
121 \PackageError{exercise}{You can't nest a 'list' inside a 'parlist'.}{%
122 I've found a 'list' or 'trivlist' environment nested inside^^J%
123 a 'parlist'. This isn't allowed. You're probably doomed now.%
124 }%
125 }
126 </package>

```

Done.

Mark Wooding, 25 August 2003

Appendix

A The GNU General Public License

The following is the text of the GNU General Public License, under the terms of which this software is distributed.

GNU GENERAL PUBLIC LICENSE

Version 2, June 1991

Copyright © 1989, 1991 Free Software Foundation, Inc.

51 Franklin Street, Fifth Floor, Boston, MA 02110-1301, USA

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software—to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation’s software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Library General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author’s protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors’ reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone’s free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The “Program”, below, refers to any such program or work, and a “work based on the Program” means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term “modification”.) Each licensee is addressed as “you”.

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program’s source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:
 - (a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.
 - (b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.
 - (c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute

them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:
 - (a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,
 - (b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,
 - (c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.
5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.
6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.
7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder

who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and “any later version”, you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM “AS IS” WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.
12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

Appendix: How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the “copyright” line and a pointer to where the full notice is found.

one line to give the program’s name and a brief idea of what it does.
Copyright (C) yyyy name of author

This program 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, write to the Free Software Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301, USA.

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

Gnomovision version 69, Copyright (C) yyyy name of author
Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type ‘show w’.
This is free software, and you are welcome to redistribute it under certain conditions; type ‘show c’ for details.

The hypothetical commands `show w` and `show c` should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than `show w` and `show c`; they could even be mouse-clicks or menu items—whatever suits your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a “copyright disclaimer” for the program, if necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the program ‘Gnomovision’ (which makes passes at compilers) written by James Hacker.

signature of Ty Coon, 1 April 1989
Ty Coon, President of Vice

This General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Library General Public License instead of this License.

Index

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

Symbols			
<code>\@addtoreset</code>	6	<code>\endexercise</code>	
<code>\@bsphack</code>	58	<code>\endlist</code>	
<code>\@enumdepth</code>	31	<code>\endparenun</code>	
<code>\@esphack</code>	66	<code>\endparlist</code>	
<code>\@ifnextchar</code>	72, 104	<code>\endtrivlist</code>	
<code>\@input</code>	72	<code>\enumerate</code>	
<code>\@itemlabel</code>	97, 104	environments:	
<code>\@listctr</code>	106	<code>doanswer</code>	
<code>\@makeother</code>	61	<code>exercise</code>	
<code>\@nmbolistfalse</code>	99	<code>parenun</code>	
<code>\@tempa</code>	24, 25, 27	<code>parlist</code>	
<code>\@trivlist</code>	96	<code>\ex@ansfile</code>	
<code>\@unexpandable@protect</code>	50	<code>\ex@ansfilename</code>	
		<code>\ex@answer</code>	
		<code>\ex@ctrcheck</code>	
		<code>\ex@ctrdo</code>	
		<code>\ex@endanswer</code>	
<code>_</code>	36	<code>\ex@ext@i</code>	
A		<code>\ex@opts</code>	
<code>\answer</code>	1, <u>55</u>	<code>\ex@withext</code>	
<code>\answers</code>	3, <u>71</u>	<code>\ex@within</code>	
<code>\answers@i</code>	72, 74	<code>\exctrcheck</code>	
<code>\answerwrite</code> ..	2, 44, <u>47</u> , 60, 63, 67, 76, 114	<code>\exercise</code>	
<code>\AtBeginDocument</code>	41	<code>exercise (environment)</code>	
B		<code>\exercisename</code>	
<code>\begin</code>	60	<code>\exerr@parlist</code>	
C		<code>\exerr@toolate</code>	
<code>\CurrentOption</code>	13	<code>\exfix</code>	
D		I	
<code>\DeclareOption</code>	13	<code>\if@filesw</code>	42, 48
<code>\do</code>	61	<code>\if@nmbolist</code>	106
<code>\doanswer</code>	89	<code>\ignore</code>	70
<code>doanswer (environment)</code>	3, <u>89</u>	<code>\ignorespaces</code>	102, 108
<code>\dospecials</code>	61	<code>\input</code>	72
E		<code>\item</code>	36, 92, 101
<code>\end</code>	67	<code>\itemindent</code>	35, 91
<code>\endoanswer</code>	94	L	
		<code>\labelsep</code>	35, 91
		<code>\leavevmode</code>	108

<code>\list</code>	110	Q	
		<code>\q@delim</code>	79, 81, 84, 85
M		R	
<code>\makelabel</code>	98, 107	<code>\refstepcounter</code>	33, 106
<code>\mkdef</code>	4	<code>\RequirePackage</code>	2, 16
<code>\mkparse</code>	12		
N		S	
<code>\newcounter</code>	15	<code>\skipanswer</code>	1, <u>70</u>
<code>\newwrite</code>	39	<code>\sv@read</code>	63
<code>\next@</code>	81, 82, 85–87	<code>\sv@readenv</code>	55
		<code>\sv@safespc</code>	62
P		T	
<code>\PackageError</code>	114, 121	<code>\textbf</code>	36, 92
<code>\parenum</code>	111	<code>\theexercise</code>	8, 9, 36, 60
<code>parenum</code> (environment)	3, <u>111</u>	<code>\toks@</code>	7, 9, 19, 20
<code>\parlist</code>	95, 110	<code>\trivlist</code>	35, 91
<code>parlist</code> (environment)	<u>95</u>		
<code>\pl@item</code>	101, <u>104</u>	U	
<code>\pl@item@i</code>	104, 105	<code>\useparlist</code>	<u>110</u> , 111
<code>\ProcessOptions</code>	14		
<code>\protect</code>	50		
<code>\providecommand</code>	30		