

*A Computer-Assisted Personalized Approach,*  
**CAPA,**  
*for Assignments, Quizzes and Examinations.*

---

*CAPA System User's Manual*  
Version 4.5

*Y. Tsai, E. Kashy, D.J. Morrissey, N.E. Davis, and G. Albertelli*  
*Michigan State University, East Lansing, MI 48824, USA.*  
June 28, 1997

Contents

<b>1</b>	<b>Installation of CAPA</b>	<b>1</b>
<b>2</b>	<b>CAPA Software Components</b>	<b>8</b>
<b>3</b>	<b>Running CAPA via X-Server software</b>	<b>13</b>
<b>4</b>	<b>Guided Tour of the CAPA System</b>	<b>17</b>
<b>5</b>	<b>Specification of CAPA Problem Sets</b>	<b>26</b>
<b>6</b>	<b>CAPA Functions</b>	<b>35</b>
<b>7</b>	<b>Auxiliary Files</b>	<b>46</b>
<b>8</b>	<b>Sample Login Instructions</b>	<b>48</b>
<b>9</b>	<b>A Problem Set with Source Code for some Questions</b>	<b>50</b>



## 1 Installation of CAPA

This Section describes how to install the CAPA software. These instructions are written for an Alpha running Digital Unix 3.2D (OSF1).

1. Log in as the super-user (root).
2. Change Directory to `/usr/sbin`
3. Enter `adduser`
4. A prompt will appear. Follow prompt to enter information about the user. The example used here is for the user name **teacher** in the group **cai**.
  - (a) Enter login name: **teacher**
  - (b) Enter a UID: For most cases, using the default UID is preferable. Just enter `<return>`
  - (c) Enter a full name for user: **Instructor for course**
  - (d) Enter a login group for user: **cai**
  - (e) Enter another group that user should be a member of.  
(`<Return>` only if none): Just enter `<return>`
  - (f) Enter a parent directory for user: `/usr/users`
  - (g) Enter a login shell for user: `/usr/bin/csh`
  - (h) Do you wish to edit the auth file entry for this user? **n**
  - (i) New password: **password**  
**NOTE:** The password will not be echoed to the screen.
  - (j) Retype new password: **password**
5. The compressed files for the CAPA system version 4.5 are to be placed in the directory `/usr/users/teacher/CAPA45`.
6. Expand these compressed files by typing `uncompress CAPA45.tar.Z`
7. Untar the directories by typing `tar -xvf CAPA45.tar` The CAPA45 directory will then contain the following items:
 

```
drwxr-xr-x  7 teacher  cai           512 May 21 13:41 .
drwxr-xr-x  3 teacher  cai           512 May 21 13:42 ..
drwxr-xr-x  4 teacher  cai           512 May 20 12:39 LinkToBin
drwxr-xr-x  5 teacher  cai           512 May 20 13:05 PutInCAPA45
drwxr-xr-x  2 teacher  cai           512 May 21 13:15 PutInClass
drwxr-xr-x  3 teacher  cai           512 May 20 12:59 PutInPublic
drwxr-xr-x  3 teacher  cai           512 May 20 12:34 PutInPublicWeb
-rw-r--r--  1 teacher  cai          3673 May 21 13:41 README.txt
-rw-r--r--  1 teacher  cai        880328 May 19 17:21 manual45.ps
```
8. Print the file name `README.txt`. Follow the instructions for installation.

9. The CAPA45.tar.Z file may be recompressed and moved to another directory if you wish.
10. After completing the items included in README.txt, check the following directories for content and permissions. File sizes may differ.

```
capa2.nssl.msu.edu> pwd
/usr/users/teacher/CAPA45
capa2.nssl.msu.edu> ls -la
total 1277
drwxr-xr-x  12 teacher  cai           512 Jun  3 07:59 .
drwxr-x--x  12 teacher  cai          1024 Jun 10 07:18 ..
drwxrwxrwx   2 teacher  cai          3584 Jun  3 22:49 MCTools
drwxrwxrwx   2 teacher  cai           512 May 20 11:20 Tools
-rwsr-xr-x   1 teacher  cai        245760 May 20 18:54 capalogin
drwxr-x--x  10 teacher  cai          1024 Jun  8 21:46 nsc111f7
```

11. Important: Note that `capalogin` should have permissions `-rwsr-xr-x`. Note specifically the `s` in the fourth position. It must be there for students to be able to login. If it is not there, enter the command: `chmod u+s capalogin`. (You may need to login as 'root' for this change to take place.)
12. Check the following directories for content and permissions. File sizes may differ.

```
capa2.nssl.msu.edu> pwd
/usr/users/teacher/PublicWeb/CAPA
capa2.nssl.msu.edu> ls -lag
total 11
drwxrwxr-x   4 teacher  cai          1024 Apr 28 14:32 .
drwxrwxr-x   3 teacher  cai          1024 Apr 28 14:32 ..
-rw-rw-r--   1 teacher  cai          2894 Apr 16 09:32 GLabel.class
drwxr-x--x   2 teacher  cai          1024 Apr 28 14:36 Icons
drwxrwxr-x   3 teacher  cai          1024 Apr 28 14:39 capa-bin
-rw-rw-r--   1 teacher  cai          1336 Apr 28 15:49 class.html
-rw-rw-r--   1 teacher  cai           254 Apr 16 09:31 help.html
```

```
capa2.nssl.msu.edu> pwd
/usr/users/teacher/PublicWeb/CAPA/capa-bin
capa2.nssl.msu.edu> ls -lag
total 679
drwxrwxr-x   3 teacher  cai          1024 Apr 28 14:39 .
drwxrwxr-x   4 teacher  cai          1024 Apr 28 14:32 ..
-rwxr-xr-x   1 teacher  cai        278528 May  8 16:57 capahtml
-rw-r--r--   1 teacher  cai           775 Jun  4 12:41 class.conf
drwxrwxr-x   2 teacher  cai          1024 Apr 28 14:39 teacher
-rw-rw-r--   1 teacher  cai        124667 Jun 10 15:33 web_access.log
```

```
capa2.nssl.msu.edu> pwd
/usr/users/teacher/PublicWeb/CAPA/capa-bin/teacher
capa2.nssl.msu.edu> ls -lag
total 282
drwxr-xr-x   2 teacher  cai           512 May  8 16:58 .
drwxr-xr-x   8 teacher  cai           512 Jun  4 12:10 ..
-rwsr-xr-x   1 teacher  cai        278528 May  8 16:57 capasbin
```

13. Important: Note that `capasbin` should have permissions `-rwsr-xr-x`. Note specifically the `s` in the fourth position. It must be there for students to be able to login. If it is not there, enter the command: `chmod u+s capasbin`. (You may need to login as 'root' for this change to take place.)
14. Check the following directories for content and permissions. File sizes may differ.

```
capa2.nsc1.msu.edu> pwd
/usr/users/teacher/Public/nsc111f7
capa2.nsc1.msu.edu> ls -lag
total 2
drwxr-xr-x  2 teacher cai      512 Mar 26 09:44 .
drwxr-xr-x  9 teacher cai      512 May 20 10:43 ..
lrwxrwxrwx  1 teacher cai      40 Mar 26 09:07 Links -> /usr/users/teacher/CAPA45/nsc111f7/Links
lrwxrwxrwx  1 teacher cai      43 Mar 26 09:44 pictures -> /usr/users/teacher/CAPA45/nsc111f7/pictures

capa2.nsc1.msu.edu> pwd
/usr/local/bin
capa2.nsc1.msu.edu>ls -la
drwxr-xr-x  7 root  system  2048 Mar 31 16:38 .
drwxr-xr-x  9 root  system  512 Mar 13 13:47 ..
lrwxrwxrwx  1 root  system  32 Mar 27 17:14 allpin -> /usr/users/teacher/CAPA45/LinkToBin/allpin
lrwxrwxrwx  1 root  system  39 Mar 27 17:14 grader -> /usr/users/teacher/CAPA45/LinkToBin/Grader/grader
lrwxrwxrwx  1 root  system  41 Mar 27 17:14 quizzer -> /usr/users/teacher/CAPA45/LinkToBin/Quizzer/quizzer
lrwxrwxrwx  1 root  system  33 Mar 27 17:14 qzparse -> /usr/users/teacher/CAPA45/LinkToBin/qzparse
```

15. Create an account for the class which will be using `CAPA`. Do this by adding the class as a separate user with a different login shell. The login name for the class is the same as the class name. In the directory `/usr/sbin`, enter `adduser`

- (a) Enter login name: `nsc111f7`
- (b) Enter a UID: Using the default UID is preferable. Just enter <return>
- (c) Enter a full name for user: `Course Name`
- (d) Enter a login group for user: `cai`  
**NOTE:** The group for the course **must** be the same as the instructor for the course. For all examples we use `teacher` and `cai`.
- (e) Enter another group that user should be a member of.  
 (<Return> only if none): Enter <return>
- (f) Enter a parent directory for user: `/usr/users/teacher/CAPA45`
- (g) Enter a login shell for user: `/usr/users/teacher/CAPA45/capalogin`
- (h) Do you wish to edit the auth file entry for this user? `n`
- (i) New password: `password`  
**NOTE:** The password will not be echoed to the screen.
- (j) Retype new password: `password`

16. The password file must now be edited to remove the class password. To do this:

- (a) `cd` to `/usr/sbin`
- (b) Enter `vipw`<sup>1</sup>
- (c) Use the arrow keys to position the cursor over the first letter of the encoded password for the class user name. For example, a typical password file entry looks like this:

```
nsc111f7:3ou7NE4pQMTw:4347:26:Natural Science College 111 Fall 1997:/usr/users /teacher/CAPA45/nsc111f7:/usr
```

Note that the fields are separated by colons. The description of the fields is as follows:  
**username:password:UID:GID:Full Name:Parent/Directory:Login/shell**

---

<sup>1</sup>vi editing is best left to users familiar with vi

- (d) Remove the password from this listing. Do this by positioning the cursor over the first letter (or character) of the password field, then delete by pressing the x key. Exit the vi editor by typing :wq The modified listing should then look like: nsc111f7::4347:26:Natural Science College 111 Fall 1995:/usr/users/teacher/CAPA45 /nsc111f7:/usr/users/teacher/CAPA45/capalgin

17. Exit the root account and login as **teacher**

18. Change directory to the sample class: `cd /usr/users/teacher/CAPA45/nsc111f7`.

19. Type: `ls -lag`

The last files in the `teacher/CAPA45/nsc111f7` directory, , will look like: (Again, file sizes will differ.)

```
drwxr-x--x 10 teacher cai      1024 Jun  8 21:46 .
drwxr-xr-x 12 teacher cai      512 Jun  3 07:59 ..
drwx----- 2 teacher cai      512 Apr 22 10:24 ASCII
drwx----- 2 teacher cai      512 Apr  4 14:41 HTML
-rw-r--r-- 1 teacher cai      769 Dec 10 13:37 HWTtop
drwxr-xr-x 2 teacher cai      512 Mar 26 09:56 Links
drwx----- 2 teacher cai      512 May 22 14:59 PIN
drwxr-xr-x 2 teacher cai      512 May 30 11:16 TeX
-rw-r--r-- 1 teacher cai        15 Dec  7 16:45 TeXfooter
-rw-r--r-- 1 teacher cai     657 Dec 12 17:11 TeXheader
-rw-r--r-- 1 teacher cai    3540 Apr 21 15:08 capa.config
-rw-r--r-- 1 teacher cai     246 Jan  8 14:28 classl
-rw-r--r-- 1 teacher cai      29 Aug 22 1994 goodbye.msg
-rw-r--r-- 1 teacher cai     777 Feb 11 1993 help.msg
drwxr-xr-x 2 teacher cai      512 Dec 10 13:52 pictures
drwxr-xr-x 2 teacher cai      512 Jan  6 12:14 records
-rw-r--r-- 1 teacher cai      362 May 22 1995 second-login.msg
-rw-r--r-- 1 teacher cai   16494 Jan  8 13:13 set1.qz
-rw-r--r-- 1 teacher cai     293 Sep 28 1994 third-login.msg
-rw-r--r-- 1 teacher cai     215 Jan  6 17:42 welcome.msg
```

**Note:** Each class added to the system must contain the records directory (created by the instructor). The `capa.config`, `classl`, `TeXheader`, and `TeXfooter` are required files. All `.msg` files contain useful information for the students and can be edited to suit the instructors' needs. The `HWTtop` is useful in formatting the assignment printed page and can also be edited. The `Links` and `pictures` directories are specific to the sample class included in this distribution.

20. Change to the records directory: `cd records`

21. Type: `ls -lag`

The directory will be empty. However, after you have run `quizzer` and students have logged in, several database files will be automatically created. The last files in this directory will eventually look like:

```
-rw-r--r-- 1 teacher cai      166 Jan  8 15:36 active.log
-rw-r--r-- 1 teacher cai     355 Jan  8 15:36 duration.db
-rw-r--r-- 1 teacher cai     294 Jan  8 15:36 log1.db
-rw-r--r-- 1 teacher cai     454 Jan  8 15:36 set1.db
```

22. Important: Check that the permissions and user-ownership of the files are the same as those in the examples above. If the group shown is not `cai`, then change the working directory to `teacher/CAPA45` by

typing: `cd /usr/users/teacher/CAPA45` and then `chgrp -R cai *` to change the group of all the files. Similarly, if the owner shown is not `teacher`, then change the working directory to `teacher/CAPA45` by typing: `cd /usr/users/teacher/CAPA45` and then `chown -R teacher *` to change the owner of all the files in the `CAPA45` directory and subdirectories.

23. Note: If you have created user IDs for classes in addition to `nsc111f7`, they will appear as subdirectories in `teacher/CAPA45/`. The ownership and group should be `teacher` and `cai` as well. The same prototype files and folders as in `teacher/CAPA45/nsc111f7` must be placed in each sub-directory ( copy and edit them as needed).

24. Some notes:

- (a) The `LATEX` and `dvi` programs should be properly installed on the host machine in order for `quizzer` to format the problem sets. See your system administrator for help in installing these programs.
- (b) `tcl 7.6` and `tk 4.2` libraries must be installed for the X-Windows version. See your system administrator for help in installing these libraries.
- (c) For a broad range of terminal emulators to be able to access the system<sup>2</sup>, it is necessary to modify the `/etc/termcap` file to accommodate the appropriate `vt100` terminal attributes, including handling non-standard nomenclature used by terminal emulation programs on Macs and PCs. A notable example is the widely used `tn` program that is part of FTP Software, Inc.'s `PC/TCP` package. This program negotiates its terminal type with the host computer using the prefix `dec-` for the `vt` family of terminals (*e.g.*, it requests terminal type `dec-vt100` instead of `vt100`). This prefix, if unknown to the Unix host, may cause the negotiation to result in an incorrect terminal type or to fail completely. The line in the standard `/etc/termcap` file that reads

```
d0|vt100|vt100-am|vt100am|dec vt100:\
```

should be edited to read:

```
d0|vt100|vt100-am|vt100am|dec-vt100|dec vt100:\
```

thereby establishing `dec-vt100` as an official synonym for `vt100`. Note that the “`dec-vt100`” entry is different from the “`dec vt100`” entry. We advise adding the new alias rather than changing the existing alias and possibly causing other problems. A similar change can be made on other operating systems which use the `/etc/termcap` method of terminal definition. Unix systems using the `terminfo` program instead will support similar minor changes to the appropriate file or files in the `/usr/lib/terminfo` directory tree (consult your system administrator for details). Be prepared to make similar changes in case other terminal emulation programs at your site use their own non-standard terminal type nomenclature.

25. The `CAPA` system should now be completely installed. You may start writing problem sets, but we recommend reading the system description and taking the guided tour.

---

<sup>2</sup>by G. Perkins, MSU P/A Dept.

The whole package for NeXTStep Operating system is essentially the same as described previously, the difference in applications is that Quizzer and Grader are named Quizzer.app and Grader.app respectively. The following items describe how to create an account on the NeXTStep Operating system, Version 3.3.

1. Log in as the super-user (root).
2. Create a new user group called **cai**. Use the menu obtained from opening the file `/NextAdmin/UserManager` in the File Viewer and double-clicking on its icon.
  - (a) select **Group**
  - (b) select **New** to create a new user group
  - (c) In the domain window, click **OK**
  - (d) type **cai** in the group name field
  - (e) select **Save** to save the user group
3. Create an Instructor account called **teacher**, using the menu of `/NextAdmin/UserManager`:
  - (a) select **User**
  - (b) select **New**
  - (c) select **Local** as user type
  - (d) select **Groups**
  - (e) select **cai** in the User Group Info window
  - (f) set default group as **cai**
  - (g) select **Long Form**
    - i. **User's Full Name:** (i.e., instructor's full name or another description, such as *CAPA Instructor*)
    - ii. **Username:** teacher
    - iii. **Password:** (enter a secret password), case sensitive
    - iv. **User ID:** (A number selected by the operating system)
    - v. **Home Directory:** `/teacher`
    - vi. **Login Shell:** `/bin/csh`
  - (h) select **Save** in the menu; follow the instructions, verify password, check values.
  - (i) select **Yes** to create a home directory, then wait for the save to be completed.
4. Create an account that will be used by the students in the class to login called **nsc111f7**
  - (a) Locate the file `/etc/shells` in the File Viewer. Double-click its icon to open the file with `Edit.app`. This will enable you to edit the file, so that the code `/teacher/CAPA45/capalogin` acts as a valid Login shell under UNIX. To complete this:
    - i. Add the line below to the `/etc/shells` file:  
`/teacher/CAPA45/capalogin`  
(with a <CR> at the end)
    - ii. select **Save** in the menu.
  - (b) Continuing with `/NextAdmin/UserManager`
    - i. select **New User**
    - ii. select **Local** as user type

- iii. select **Groups**
  - iv. select **cai** in the User Group Info window
  - v. set default group as **cai**
  - vi. select **Long Form**
    - A. **User's Full Name:** ( i.e., Natural Science Students Fall 1997)
    - B. **Username:** *classname* (here enter: nsc111f7)
    - C. **Password:** (none, just hit return)
    - D. **User ID:** (a number selected by the operating system)
    - E. **Home Directory:** *teacher sub-directory* (here enter : /teacher/CAPA45/nsc111f7)
    - F. **Login Shell:** /teacher/CAPA45/capalogin (this specification is the one that limits student access)
  - vii. select **Save** in the menu; a warning message may appear
  - viii. select **OK**, check values.
  - ix. To the prompt: **Create User Home Directory**, say **No**.  
This is very important; Students do not get file space.
  - x. **Quit** UserManager
5. Return to the file `/etc/shells`
  6. Edit the file by removing the line  
`/teacher/CAPA45/capalogin`  
Note: This allows students to login but only under the restrictions placed by the `capalogin` code. Otherwise, students (i.e., user *nsc111f7*) could login on your computer and see or modify all the files in teacher that have group read/write permission.
  7. select **File/Save**
  8. select **File/Close**
  9. **Quit** Edit
  10. Several **Username** sub-directories which students use to login and that hold all of the necessary files can be placed under the teacher account, just as was done for nsc111f7 above, i.e., phy232f7, cem152f7, xph232f7, etc.
  11. **logout** as root.



---

## 2 CAPA Software Components

quizzer, qzparse, capalogin, capasbin, capahtml, grader and their auxiliary files

---

There are six application codes required for the complete system. They are: `quizzer`, `capalogin`, `capasbin`, `capahtml`, `qzparse` and `grader`. The `capalogin` code is executed automatically for each student session, the `quizzer` and `grader` applications are tools for the instructor to manage the problem set files and the database, the `qzparse` code is used to print problem sets in batch mode after they have been created. Similar in functionality to `capalogin`, both `capahtml` and `capasbin` are programs executed by web server when a student uses web browser to answer the problem set.

### 1. quizzer

- **Location:** The `quizzer` application can reside in any directory and can be linked to a directory within the users' path. For example, linking `quizzer` to `/usr/local/bin` would make the application available to all users on the machine.
- **Function:** The major functions of the `quizzer` application are:
  - Create new problem-set files, named `set1.qz`, `set2.qz` ... , etc. containing ASCII text only.
  - Edit existing question files using ASCII text only.
  - Calculate and preview the answers generated by the problem set code.
  - Preview the resulting problem set (which will be typeset with  $\text{\LaTeX}$ ).
  - Preview the resulting problem set as would be seen by students when they log in through a terminal window session. (Enscript Mode)
  - Preview the resulting problem set showing the coded HTML flags.
  - Generate the database file (`setx.db`) which contains the time constraints on the availability of the problem sets to students for a given assignment. Each problem set has a opening-date, a due-date, and a date when answers are available.
  - Print the problem sets for specific students with the CAPA IDs (CAPA Identification Numbers) needed to login. The CAPA ID identifies the set and we recommend printing this number on student papers.
  - Print the problem sets for a specific section.
  - Print the problem sets for the whole class (If the class is very large, we recommend printing the sets in batch mode.)
  - Print answer-only papers to facilitate the hand-grading of CAPA-based examinations.<sup>3</sup>
- **Files:** `set1.qz`, `TeXheader`, `TeXfooter`, `capa.config`, and `class1`.
  - Problem set files are the files containing the coded problems created by instructor for each given assignment. They are labeled `set1.qz`, `set2.qz`, ... etc. The problem sets must start with number 1 and continue in increasing numerical order without omissions. There is a fifty question limit within each problem set.
  - `TeXheader` and `TeXfooter` specify the information needed by the  $\text{\LaTeX}$  formatter to create the top and bottom of the document, respectively. The `nsc111f7` example file contains copies of the `TeXheader` and `TeXfooter` files which you can copy to your class and modify as necessary.

---

<sup>3</sup>An automated scanning system for personalized examinations and quizzes has also been developed.)

- `capa.config` is the unit configuration file which contains the standard SI units with all prefixes. It also contains a list of derived SI units. The file may be edited (maintaining file format) to include other units at the instructors descretion.
- `class1` is the class list file for each specific class. The maximum number of students in one class is 4096. The format of the entries in the class list file is strict and must correspond exactly to that in the following example.

\* The sample file, `class1`, with the information for enrolled students must have the following structure:

```
123456789012345678901234567890123456789012345678901234567890123456789012345
nsc 111 001 A12345678 Davis, Nancy . ndavis@nscl.msu.edu
nsc 111 001 A87654321 Student, Jamie .
nsc 111 001 173336318 Kashy, Edwin N. kashy@nscl.msu.edu
nsc 111 002 A51413121 Tsai, Isaac Y.
nsc 111 002 173336321 Morrissey, David J.
nsc 111 002 173336323 Thoennesen, Mike .
```

\* Note: The first row of digits is **NOT** part of the file but is given here to locate the information in the file.

\* The fields used by this system are:

- (a) Section number which appears in columns 11 to 13
- (b) Student Number in columns 15 to 23
- (c) Student Name beginning in column 25.

The maximum number of characters in the Student Name field is 30 at present. The course acronym and number (in columns 1 to 10) are not used by the system, but are convenient for recordkeeping. The format of the file must be exact. The section number is used for printing in batch mode.

- \* Each line, including the last line, must end with a carriage return. There should *NO* blank lines in the file. There should also not be any extra empty lines at the end; they may cause multiple printing of the problems for the last (real) student in the list.
- \* After line character number 56, any comment can be added. We have used this space for e-mail addresses for contacting students.
- \* The period (.) added for students with no middle names has facilitated importing CAPA summaries into some spread sheet programs (i.e., Lotus) when preparing final grades for a course.

- **records** directory:

- This folder must be present as a sub-directory of the class directory. `quizzer` saves the date information for `setx.qz` in `records/setx.db`. `capalogin` saves student interaction for the telnet session for `setx` in `records/logx.db`. `capasbin` saves student interaction for the web session for `setx` in `records/weblogx.db`. The summary information from both `logx.db` and `weblogx.db` is written to the `setx.db`. The duration of student sessions is saved in the file `records/duration.db`. As students login, a file named `active.log` is created. It is used to limit the number of concurrent telnet sessions per student.

## 2. qzparse

- **Functions:** `qzparse` is used to generate the `.tex` files needed to prepare problem sets in a batch mode without using `quizzer`. It can create *one* output file for an entire section or a multiple number of sets for a given student. The output file can then be processed by `LATEX` and `dvips` to produce a postscript file for printing. The options of `qzparse` are displayed by typing `qzparse -h`. Examples:

```

capa1.nscl.msu.edu> qzparse -h
USAGE: qzparse [ -[T|H|A][a|b] ] [-Sec [n|n:m] | -Stu sn] [ -Set [n|n:m] ]
  Example 1: qzparse -Tb -sec 2:3 -set 2:5
             will generate tex files with both questions and answers
             for sections 2 to 3, sets 2 to 5
  Example 2: qzparse -Ha -stu A12345678 -set 3
             will generate html files with answer only
             for student A12345678 set 3
  -T      = tex    mode
  -H      = html  mode
  -A      = ascii mode
           = default question only
  a      = answer  only
  b      = both question and answer
  -Sec 3  = for section 3
  -Sec 3:7 = from section 3 to section 7
  -Stu A12345678 = for a specified student
  -Set 1  = output set 1
  -Set 3:4 = output from set 3 to set 4
  -o output_filename_with_absolute_path (only for a student)
-----This is version 4.5 @ Feb 11, 1997

```

- (a) For preparing TeX files for set 4 papers for all students in sections 1 thru 45:  
`qzparse -T -sec 1:45 -set 4`
- (b) For preparing papers which only contain the answers of a particular set for all students in section 3.  
`qzparse -Ta -sec 3 -set 4`
- (c) For preparing set 1 thru 5 for a student with student number A87654321.  
`qzparse -T -stu A87654321 -set 1:5`
- **Files:** `qzparse` uses the same files as `quizzer`. In addition, running `qzparse` will create a TeX sub-directory in the class directory to house the `qzparse` output files, i.e., `section1-set1.tex`, `a87654321.tex`, etc. The `.tex` files must be passed through  $\text{\LaTeX}$  and the `dvips` codes to make postscript files for printing. (See guided tour of `qzparse` for explanations on how this is accomplished.)

### 3. `capalogin`

- **Function:** The main function of the `capalogin` code is to handle remote sessions of the students who access *CAPA* with VT100 terminals. The code is run instead of a UNIX shell in order to allow large numbers of students to easily login while controlling their access to the data files. `capalogin` checks if the student is already logged-on in this class. If the student is already logged in, then a warning message is sent (see below) and the student is allowed in to the system. If the student has opened 2 sessions without exiting properly, the student is not allowed into the system and is sent a different warning message.
- **Files:** The instructor can send general information to the entire class by typing messages into certain files. The login-specific files, `welcome` and `goodbye`, (and if needed the `help`, `second-login.msg`, and `third-login.msg`), can be edited to send information to students, reminding them of deadlines or of exam dates, telling them to disregard problem xx (when a serious error has been made in coding), etc... Note that the instructor can code hints and explanations for individual problems directly in the `setx.qz` files. Files and their functions are listed below:

<code>welcome.msg</code>	⇒ The message displayed to the student upon login.
<code>help.msg</code>	⇒ The message displayed when the student selects the menu item <code>Help</code> .
<code>goodbye.msg</code>	⇒ The message displayed after a student selects menu item <code>eXit</code> .
<code>second-login.msg</code>	⇒ The message displayed when a student already is logged on and tries to begin a second session.
<code>third-login.msg</code>	⇒ The message displayed when a student already is logged on twice and tries to begin a third session.
<code>Exam.path</code>	⇒ <i>Optional</i> , contains the absolute path to the Exam class folder (if one exists). A student can access exam scores from the Main Menu.
<code>Quiz.path</code>	⇒ <i>Optional</i> , contains the absolute path to the Quiz class folder (if one exists). A student can access quiz scores from the Main Menu.

#### 4. `capahtml` and `capasbin`

- **Functions:** The major functions of `capahtml` are to:
  - Authenticate the student number and CAPA ID entered by the student and generate the main menu page after verifying them.
  - Produce the page that contains the corresponding problem set when a student selects “Try current set” button from the main menu page.
  - Display a summary of student grades when “Term summary” button is selected.
- **Functions:** The major functions of `capasbin` are to:
  - Check the correctness of answers submitted by the student when clicking “Submit All Answers” button.
  - Give the appropriate response to a student’s entry and display the correct answer when the entry is satisfactory.

#### 5. `grader`

- **Functions:** The major functions of `grader` are to:
  - Generate grade reports for a student, a section, or for the entire class.
  - Facilitate grading and recording of individual assignments that are not entered remotely via `capalogin` or `capaweb`. `grader` displays the range of ‘correct’ answers for each specific student’s assignment and the instructor can hand-grade and update records for any assignment hard copy turned in by a student.
  - Compile the record of logins for a particular student.
  - Provide a ‘Synopsis’ or on-line summary of all the students performance on the questions within a set.
  - Display and thus provide a check of the open/close dates for any problem set.
  - Display a summary of student grades and CAPA ID numbers for any problem set.
- **Files:** `grader` reads the `class1` file and the files in the `records` directory generated by `quizzer` and `capalogin` and/or `capaweb`. The reports created by `grader` are put into the class directory; the reports for the entire class are labeled `ClassSet1.rpt` and reports for a section are labeled `Sec1Set1.rpt`, etc. They should be renamed if permanent records are required, i.e., a later `ClassSet1.rpt` file will overwrite any previous `ClassSet1.rpt`.

6. **Tools**, **MCTools**, .. The **Tools** directory contains ASCII files of macros developed to simplify the coding in the preparation of assignments and are self-documenting. The functions that they perform can be coded into the `setx.qz` files directly. These macros are provided for your convenience. The **MCTools** directory provides pre-assembled templates that simplify the coding of conceptual problems in various formats, or where ranking on a numerical criterion is desired.

---

### 3 Running CAPA via X-Server software

---

Your current computer system can be used to access the applications and edit your files on the CAPA host. This section describes examples for using a PC or a Macintosh computer to edit files and run the **quizzer** application.

- **Running CAPA via eXceed**<sup>4</sup>

**NOTE:** These instructions are for using eXceed Version 4.0 from a PC running Windows-NT<sup>5</sup> in connection with a DEC Alpha host running OSF1.

**Procedure**

- To set up eXceed to automatically open upon start-up of your PC:
  1. Install the software following the directions from the producer.
  2. Open the path `C:/Winnt/Profiles/Username/Start Menu/Programs/Startup` from 'My Computer'.
  3. Open the path `C:/Winnt/Profiles/Username/Start Menu/Programs`
  4. Highlight the `eXceed.exe` by single clicking on it. From the menu bar choose File, then Create Shortcut.
  5. Drag the Shortcut you have just created into the Startup folder you opened previously.
  6. When you reboot your machine, eXceed will execute upon startup.
- To set up your CAPA server as an xhost: Two methods:
  1. The simplest way is to leave the `host.txt` file in eXceed as deactivated. This allows connection with any host as X-server.
  2. If you wish, you can specify each host by creating the `host.txt` file in eXceed.
    - (a) From the Windows-NT main menu bar, choose Start, choose Programs, choose Exceed, then choose X-Config.
    - (b) Double click on the Security icon.
    - (c) A pop-up menu will appear where you can specify using the `host.txt` file by clicking on the associated radio button.
    - (d) You can choose to edit this file and add your hostname. Remember to save the `host.txt` file after you have finished editing it.
    - (e) This allows your machine to connect with only the hosts listed in the `host.txt` file.
- To run a telnet session through eXceed:
  1. From the Windows-NT main menu bar, choose Start, choose Programs, choose Exceed, choose htelnet. From the menu bar, choose File, then Create Shortcut. Drag the Shortcut you have just created to the Desktop.

---

<sup>4</sup> eXceed © Hummingbird Communications, Ltd.

<sup>5</sup> Windows-NT © Microsoft Corporation

2. Double click on the shortcut you have just created to open a telnet session into your host machine.
  3. The Connect pop-up screen appears. In the field "host", enter your host name. Example: `capa1.nsc1.msu.edu`
  4. Select OK.
  5. You will then see your Telnet window with the `login:` prompt displayed.
  6. Login to your host as you normally would for a telnet session.
  7. You must then command your host to display to your local machine by entering the command `setenv DISPLAY 35.8.33.131:0.0`  
**NOTE:** Enter the IP address of the machine you are using. For this example, 35.8.33.131 is used. You can, if you wish, create an alias for this command by editing your `.cshrc` file in your login directory.
  8. You can now change directory to the application you wish to run and enter the command for running the application. If you have linked the application to a directory specified in `PATH`, then you can run the application from any directory by entering `quizzer`.  
Example: `cd CAPA45/Quizzer`  
`quizzer`
  9. The `quizzer` main menu will appear in the upper left-hand corner of your screen. You can edit and preview your files through the `quizzer` application. Be certain to quit the `quizzer` application when you are finished editing.
- To map the shortcut keys for `quizzer`:
1. From the Windows-NT main menu bar, choose Start, choose Programs, choose Exceed, then choose X-Config.
  2. Double click on the Input icon.
  3. There will be displayed a pull-down menu to set the choices for the Alt key.
  4. You can choose to have either the left or right Alt key function for your shortcut key in `quizzer` and `grader`.

- **Running CAPA via X-WIN32<sup>6</sup>**

**NOTE:** These instructions are for using X-WIN32 in connection with a DEC Alpha host running OSF1.

**Procedure**

- To set up X-WIN32 to automatically open upon start-up of your PC:
1. Install the software following the directions from the producer.
  2. Once the software has completed its installation, a small window appears which contains two shortcut icons.
  3. Open the path `C:/Winnt/Profiles/Username/Start Menu/Programs/Startup` from 'My Computer'.
  4. Drag the shortcuts into the Startup folder.
  5. When you reboot your machine, X-WIN32 will execute upon startup.
- From the toolbar, choose X-Win (blue), choose Sessions, then choose XDMCP-broadcast. This will generate a listing of all machines able to run XDMCP mode.

---

<sup>6</sup> X-WIN32 © StarNet Communications

- Open the X-Win Util (green), choose Edit Sessions, choose New Session.
- Enter the name you wish to use for your session. Example: `capa1`
- Select XDMCP connect mode. Enter the hostname. Example: `capa1.nsc1.msu.edu`
- Select OK
 

You can now run your X-Server sessions by clicking on X-Win (blue), choosing Sessions, and choosing the session name you have just created. Your screen should then appear exactly as if you were sitting at the monitor of your host machine with your Windows-NT desktop in the background.
- Change directory to `teacher/CAPA45/Quizzer`
- Enter `quizzer`
- The `quizzer` main menu will appear in the upper left-hand corner of your screen. You can edit and preview your files through the `quizzer` application. Be certain to quit the `quizzer` application when you are finished editing.

### • Running CAPA via MacX<sup>7</sup>

**NOTE:** These instructions are for using MacX Version 1.2 in connection with a DEC Alpha host running OSF1.

#### Procedure

- To open an xterm session into the server:
  1. Open MacX by double clicking on the MacX Icon. This will automatically open a document called "Untitled".
  2. From the menu bar, choose Remote then choose New Command.
  3. A New Remote Command Screen will pop up.
  4. In the field "Remote Command" enter
 

```
/usr/bin/X11/xterm -display 35.8.33.61:0.0
```

**NOTE:** Enter the IP address of the machine you are using. For this example, 35.8.33.61 is used.
  5. In the field "Command Name", enter the name you want your command to be called. Example: `capa1`
  6. In the field "Display", choose Color Rootless.
  7. In the field "Output", choose Save.
  8. In the field "Username", enter your username for access to your host. Example: `teacher`
  9. In the field "Password", enter your password for access to your host.
  10. Select Host. Another pop-up screen will appear.
  11. In the field "Method", choose MacTCP Tool.
  12. In the field "Host name or address", enter your hostname. Example: `capa1.nsc1.msu.edu`
  13. Select OK. The second pop-up window will disappear.
  14. Choose Set. The original pop-up window will disappear.
  15. From the menu bar, choose Remote. Select the new command which should show as you have named it, on the bottom of the choicelist under Remote.

---

<sup>7</sup> MacX ©1995 Apple Computer, Inc. All rights reserved.



16. You will be prompted to enter your password; select OK. The next window that opens will be your xterminal into your host. You can now use your account on your host through the xterm application. Be certain to logoff from your host (by typing `exit`) to close the xterm application when you are finished.
- To set up direct access to `quizzer` and/or `grader`.
1. From the menu bar, choose Remote then choose New Command.
  2. A New Remote Command Screen will pop up.
  3. In the field “Remote Command” enter

```
source /etc/csh.login;source .cshrc;setenv DISPLAY 35.8.33.61:0.0;
cd /usr/users/teacher/CAPA45/Quizzer;quizzer
```

**NOTE:** Enter the IP address of the machine you are using. For this example, 35.8.33.61 is used.
  4. In the field “Command Name”, enter the name you want your command to be called. Example: `quizzer`
  5. In the field “Display”, choose Color Rootless.
  6. In the field “Output”, choose Save.
  7. In the field “Username”, enter your username for access to your host. Example: `teacher`
  8. In the field “Password”, enter your password for access to your host.
  9. Select Host. Another pop-up screen will appear.
  10. In the field “Method”, choose MacTCP Tool.
  11. In the field “Host name or address”, enter your hostname. Example: `capa1.nsc1.msu.edu`
  12. Select OK. The second pop-up window will disappear.
  13. Choose Set. The original pop-up window will disappear.
  14. From the menu bar, choose Remote. Select the new command which should show as you have named it, on the bottom of the choicelist under Remote.
  15. You will be prompted to enter your password; select OK. The `quizzer` main menu will appear in the upper left-hand corner of your screen. You can then edit and preview your files through the `quizzer` application. Be certain to quit the `quizzer` application when you are finished editing.

---

## 4 Guided Tour of the CAPA System

---

This section presents a guided tour through the CAPA software system by relying on the files that are part of the distribution package.



### 1. Guided Tour of quizzer

- Instructions for running the `quizzer` application:
  - Prior to opening the `quizzer` application, set up your X-server software to interact with your server. See the section on using CAPA via eXceed, X-win32 or MacX.
  - Through your X-server software, open an rlogin (or telnet) session to your server.
  - Change directory to the directory in which the `quizzer` application is located.  
For example: `cd CAPA45/Quizzer`
  - Start the application by entering the command `quizzer`.
- Select the **Source** of the problem set.
  - The `quizzer` main menu will be placed in the upper left corner of your screen.
  - Select **File**, then **open**.
  - A pop-up screen will display your current directory. The listing on the left side is the directories within your current directory. The directory listing which consists of two dots (`..`) is the path to the previous directory. The listing on the right side of the pop-up screen is the files in your current directory which have been filtered by the conditions listed in the top line of the pop-up window. The file filter, which for `quizzer` will read `*.qz`, selects only `setx.qz` files for editing with the `quizzer` application.
  - Find the sample class file `nsc111f7`.
  - Choose `set1.qz` files and click on the OK button. (`set1.qz` is identical to the sample assignment presented on page XXX of this manual. Note that `set2.qz` has the same problems as `set1.qz` but is assembled using a library format. )
  - The contents of this file will show up in `quizzer`'s editing window. Browse through the contents of this question file. Be careful not to make any changes at this time, or at least not to **save** any changes. If you make an inadvertent change or if you are not sure of a change, then just quit the `quizzer` application and restart it.
- Preview the example question file in `enscript` mode.
  - Choose `enscript` mode from the options at the top of the `quizzer` edit window.
  - Click the **Preview** button. A dialog window will appear to ask you to select a specific student from the class list. The choice may be made randomly from a specified section or the student number of a student in a section can be entered.
  - Select the random-selection option and click on the OK button.
  - Note: To select a specific student, first, click the **Preview** button and then type the appropriate Student Number into the text entry panel *followed by a carriage return*. If the Student number is not in the class list files, (`teacher/CAPA45/nsc111f7/class1`), an error message is returned.

- The **prompt** window which appears indicating the number of questions has changed is discussed later, for now, choose **Ok**.
- A preview window will also appear. This window displays the text from the translated question file as it would be presented to the students during a terminal session. The instructor can use this display to determine if the coding is correct and if the presentation on the remote terminal will be adequate. (If any changes have been made to the `setx.qz` file in the editor, **quizzer** will ask you to save the file. Don't save the file at this point unless you are absolutely sure that the file has not been damaged, instead select **cancel**, quit **quizzer**, and start again.)
- Preview the example question file in  $\text{\TeX}$  mode.
  - Close the preview window by clicking on the **Dismiss** button at the top of the window.
  - Change the **Mode** of the preview output by clicking on the  $\text{\TeX}$  mode button on the upper part of edit window and then click the **Preview** button as before.
  - Select a particular student, this time type the student number **A87654321** into the text panel, then `<return>`, and click **OK**. The preview window will display the contents of the `nsc111f7/quiztemp.tex` file that will be used later by  $\text{\LaTeX}$  to typeset the problem set for the selected student, in this case 'Jamie Student'. (Note: For the NeXT version, these files are named `sample.tex`)
- Specify the date limits/database header.
  - The database header for `set1.qz` may already be present from the distribution. However, you can bring up a window to modify that information by clicking on the **DB header** button at the top of the main edit window. NOTE: You must preview the set prior to changing the DB Header. Three dates must be set:
    - (a) **Open date/time**, when the students can begin to enter answers to this problem set.
    - (b) **Close date/time**, the time by which students must finish entering answers and properly log out of *CAPA*, i.e., when all answer-recording stops.
    - (c) **Answer date/time**, when the answers to the problems will be made available to the students. This should be *after* the Close date/time.
  - Use the **Load** button to view the current values for this problem set. You receive a warning if the values are not present in the `set1.db` file.
  - Setting the **DB Header** is necessary for each new problem set before students login. Without the proper date information, students will not be allowed to enter their answers.
    - (a) The database header may be set as the number of problems change, but it *must* be set *after* the number of problems in the set is finalized, otherwise serious errors in the grade records will occur. If you have found it necessary to make major changes in your source file [such as changing the problem value or the grading option to Hand Graded, HGR-on in `/ANS()`], be certain to reset the **DB Header** prior to printing the final version for distribution to your students.
    - (b) Once the database header has been set and the problem set distributed to students, do *not* change the number of problems in the set and reset the database header, as this will cause the student records to be corrupted each time a student logs in.
    - (c) Resetting the database header with new dates and problem weights, will not corrupt the student records *as long as* the number of questions has not changed.
  - The dates and times must be entered with the numerical format of `MM/DD/YY` and `HH:MM` on the appropriate lines. The hours of the day are based on a 24 hour clock.
  - This panel has a cancel button to exit the window without changing the dates already present in the records folder.

- Generate quiztemp.dvi file necessary for printing.
  - Having successfully previewed the questions from the `set1.qz` file in TeX format, select **Create .dvi** from the main **quizzer** menu. (The creation of a \*.dvi file is a necessary step before printing any TeX document.) There are no **quizzer** errors in the original example file, and **quizzer** will have generated a file called `quiztemp.tex` (Note: In the NeXT version, this is named `sample.tex`) in the previous steps including everything needed to typeset the question file. After selecting the **Create .dvi** option, you will again be prompted to select either a random student or a specific student determined by student number. **quizzer** invokes the L<sup>A</sup>T<sub>E</sub>X program to typeset the `quiztemp.tex` file.
  - There are no L<sup>A</sup>T<sub>E</sub>X syntax errors in the example file so **quizzer** will create the `quiztemp.dvi` file. The `quiztemp.dvi` file is automatically opened and displayed in the X-windows version once it is created. This process takes only a few seconds to complete depending on the computer. All of the `quiztemp.*` files are in the `nsc111f7/` directory.
  - Note that new problem sets may have coding errors in their CAPA grammar, in their L<sup>A</sup>T<sub>E</sub>X grammar, in their HTML grammar, or in all three. These are independent languages and so the steps (Enscript and TeX mode Preview, and browser display) in the document preparation are necessary to detect each kind of error.
- Printing the example problem set through **quizzer**.
  - From the main **quizzer** menu, choose Print.
  - The pop-up screen will indicate your options. Choose:
    - (a) **Print current .dvi** This option will print the most recent .dvi file you have generated using the **Create .dvi** menu option. The filename you will print by choosing this option is the current version of the `quiztemp.dvi` file listed in your directory.
    - (b) **Randomly select one student from section --**. This will generate a random assignment to be printed.
    - (c) **Specify the student by:** Student Number. Once the student number is entered into the field, be certain to press <return> to call the student name from the classl.
    - (d) **Print section --** This will print an entire section's papers.
    - (e) **Print whole class** This option will print all the papers for the entire class.
    - (f) **NOTE:** For the X-Windows version, once you have entered a command for printing, another pop-up screen will appear requesting entry of the options to the `lpr` command. The required options here will be a printer specification, i.e., `-Pprintername`. Other options which can be entered here can be found by searching the Unix manual pages through a line command on your server. The manual page can be read by entering `man lpr` during a regular telnet session into your server.
    - (g) After the printing has finished, note the CAPA ID number at the top of the printed page.
  - Preview the example question file in Web mode.
    - \* This mode will generate a preview window displaying the web specific version of your source file. (This is not an .html file.) If you have not specified any functions for the web separately from those for the ASCII version (Enscript), then this preview will be identical to the Enscript preview. Web specific functions are described in the Table of Intrinsic Functions.
  - Other **quizzer** main menu items:
    - \* All **quizzer** submenus can be “torn off”. The submenus have a dashed line at the top. If you click on the dashed line, the submenu becomes a separate, static window which can be positioned anywhere on your screen.

- \* **File:** This was discussed above, but also includes “Open Reference File” a useful option allowing copying and pasting from any file into the quiz set.
  - \* **Edit:** This submenu contains the standard editing commands you will find useful.
  - \* **Find:** This submenu allows you to call up either a find panel or a line range window. The find panel allows you to find and/or replace using several different criteria. The line range window allows you to search for a specific line and character number within that line.
  - \* **Preferences:** This feature can be used to choose what style of output is displayed; whether you wish to have just problems, problems and answers or answers only displayed. This also determines what is printed on the paper through the print command, so be certain to set it to “problems only” prior to printing the entire classes papers. You can also edit the TeXheader and TeXfooter files through the preferences window. For most applications of CAPA however, you should not need to edit these files.
  - \* **Windows:** This submenu displays all the currently open windows in the `quizzier` application. If ever a smaller window is hidden by a larger one, you can shuffle the smaller window to the top by selecting it in this menu.
  - \* **Remap:** You have the option of remapping either the backspace or the delete key to suit your particular preference.
- Select **Quit** from the menu to leave the `quizzier` program.
  - A **MakeSure** window will appear. You have the option to continue editing or to quit `quizzier`.
  - The `quizzier` main menu then disappears.

## 2. Guided Tour of `capalogin`

After the instructor has created the question file (`set1.qz` in this example) and the `records/set1.db` file containing the relevant dates and times, students can login through the network. The students can use equipment or a communication program that emulates a `vt100` terminal to enter their answers and view the hints. Note that the emulation is very specifically `vt100` and not `vt102`, `vt200`, etc. See the installation instructions for information on making additions to the recognized terminal emulation on page 5.

At Michigan State University, the students login to the system from a large number of remote locations on and off campus. They have used a variety of computers, IBM, Mac, etc., that emulate a `vt100` terminal. For the guided tour, you will shortly open a terminal window on the host computer.

- **Login Instructions:** An example of a login instruction handout which has been used at MSU is reproduced in this manual on page 8.
- A student logs in by entering the class account username, in our example, `nsc111f7`. This account does not have a password so anyone can get in, but they are ‘captured’ by the code `capalogin` and have limited access to the machine (see details on page 3).
- The student is then asked to enter his or her student number and the CAPA ID number of any assignment.
  - The Student Number has 9 characters. At MSU the last 8 are digits, but the 9 characters can be a mixture of digits and letters. The CAPA ID number is unique for each problem set for a given student in a given course. It has 4 digits, and will be printed on a student’s individual assignment sheet.
  - Each student has a different CAPA ID number for each problem set. The `capalogin` code will select the `setx.qz` file that corresponds to the CAPA ID. Previous sets can be reviewed anytime.

- If the current date and time are within the allowed time range specified in the DB Header file, students can enter their answers. The answers are recorded as they are submitted.
- Problems arise when students open multiple sessions. If two sessions for one student are active at the same time, the computer records the distribution of correct and incorrect answers of the last session to log-out, not necessarily the session with the most credit earned. This is the reason for limiting active sessions.

- **Sample login:[A]**

- Open a new telnet (or rlogin) session into the server. The method for this varies depending upon the type of X-server software being used. (See Section 3 on configuration of X-server software.)
- At the prompt `login:`, enter the classname. Example: `nsc111f7`
- The `capalogin` screen will be displayed.
- Enter: `A87654321` for the student number. Note that the cursor should be positioned after the colon following the words **Student Number** near the middle of the screen. If it is not, particularly if it is at the bottom of the screen, the terminal emulation is not VT100 and needs to be reset in the terminal emulator.
- Enter the CAPA ID as was printed on the homework page for student number A87654321. The CAPA ID number must correspond to the one printed on the problem set for this student, or the login will not be allowed.
- The student's name is displayed at the top of the screen and the main *CAPA* menu near the center.
- Enter: `S` (or `s`, as the menu is not case sensitive) to view the student's summary for the course. No credit should have yet been earned.
- Enter: `M`, to return to the main menu.
- Enter: `T` to try the problem set.
- Enter: `1` to try problem 1.
- Problem 1 of set 1 will appear on the screen.
- Follow the instructions, noting that if you put in a wrong answer, a `:H` to receive a hint may become an option. Note that the colon in this screen differentiates a command from a problem answer.
- After answering a few questions, Enter: `:m` to return to the main menu.
- Enter: `s` You can now see the updated student summary.
- Enter: `x` to exit. This ends the session, records the student responses, and closes the connection to the server. The `goodbye.msg` file is then displayed .

- **Sample login:[B]**

- Open a web browser session into the server.
- Select the sample class `nsc111f7` from the pop-up menu.
- Enter: `A87654321` for the student number. Note that you must click in the window.
- Enter the CAPA ID as was printed on the homework page for student number A87654321. The CAPA ID number must correspond to the one printed on the problem set for this student, or the login will not be allowed.
- The next document you will see will be the "Main Menu" where you can select to try the current problem set.
- Set 1 will appear on the screen.

- Note that if you put in a wrong answer, a hint will be displayed if it has been included in the code.
- Note that the problem numbers are listed across the page at the top and bottom of the set. You can click on the problem number and go directly to that problem.
- Once you're finished previewing this set, you can choose to "Exit" the system.



### 3. Guided Tour of grader

- Prior to opening the `grader` application, set up your X-server software to interact with your server. See Section 3 on using CAPA via eXceed, X-win32 or MacX.
- Through your X-server software, open an rlogin (or telnet) session to your server.
- Change directory to the directory in which the `grader` application is located. For example: `cd CAPA45/Grader`
- Start the application by entering the command `grader`.
- The `grader` main menu will appear in the upper left corner of your screen.
- Choose `File` then `Specify Class`
- A pop-up window will appear to let you specify the class directory (in this case, `nsc111f7`). You will probably need to go back one directory (by double-clicking the `..`) Highlight the `nsc111f7` directory, then click `OK`. `nsc111f7` now appears in the Title Bar near the top of the `grader` window.
- Specify a section number and problem set number: Type the section number in the text entry panel and enter with `<return>`. Use section number 1 for the present example. Enter the number of the problem set that you would like displayed in the panel below. Use problem set number 1 for this example. Now, click on the `Load Scores` button. `grader` will scan the record files and the (short) list of students in section 1 and their current grades will appear. Note, for example, the grade of student `A12345678`. For classes with large student enrollments and long problem sets, the display of the section grades may take a few seconds. It is recommended that you divide large lecture groups into smaller sections.
- Select a student to grade: All the students in the selected section are contained in the displayed list. Select a student by clicking and highlighting a student's name. Click the `Grading` button. A new grading window will appear on the screen. The answers to the problem set for that student are displayed along with rows of buttons. Note that the `Answer only` option has been selected. There is also an option to view the text of the questions along with the answers. You can change the recorded grade for any problem by clicking on the corresponding radio button on the left hand side. The buttons refer to whether the answer was remotely entered as correct `Y-es`: as incorrect `N-o`: or excused `E-xcused`. A dash indicates that the problem was never attempted. Lower case `y` or `n` are written in the `setx.db` file to indicate when a problem was graded by teaching staff using `grader`. Problems whose answers cannot be entered by computer are Hand-graded problems. Essay questions or derivations are examples. (As seen later, they are identified as such in the answer format of the problem as `HGR=on`) To assign a grade, click on the box that is presented, assign the number of points received by the student in the panel that opens, and click to enter the grade.
- Save result: After changing a student's grade, click on the `Save` button (upper part of the grading window) to record the result. Confirmation of any grade change is required in a separate panel.
- Generate reports: Two types of reports can be generated by the `grader` program, a class report and a section report. Both types are saved in a text file with an `.rpt` extension in the course sub-directory, `nsc111f7`. These files will have a prefix `Class` or `Sec#`, respectively, joined with the problem set number. To generate reports select `Create class summary` from the `File` menu in the main `grader` menu and follow the instructions.
- Quit: Select `Quit` in the main menu to exit the `grader` program.



#### 4. Guided Tour of qzparse

Open an `rlogin` (or `telnet`) session as `teacher` and follow the script below.

The computer used in this example has a prompt `capa2.nsl.msue.edu>`

- `capa2.nsl.msue.edu> cd CAPA45`  
This changes the directory to `teacher/CAPA45`
- `capa2.nsl.msue.edu> qzparse -T -sec 1:2 -set 1`  
qzparse running in TeX mode, question only, from section 1 to 2, set 1 Enter the ABSOLUTE path of class  
`/usr/users/teacher/CAPA45/nsc111f7`

```
Section 1: 3 students
  Student: Davis, Nancy .           A12345678   set 1.....
  Student: Student, Jamie .        A87654321   set 1.....
  Student: Kashy, Edwin N          173336318   set 1.....
DONE set 1
```

```
DONE section 1
Section 2: 3 students
  Student: Tsai, Isaac Y.           A51413121   set 1.....
  Student: Morrissey, David J.      173336321   set 1.....
  Student: Thoennessen, Mike        173336323   set 1.....
DONE set 1
```

```
DONE section 2
ALL DONE
```

The above message shows that the `qzparse` application has generated the appropriate `.tex` files from the quiz set files. The `.tex` files will contain questions for `.qz set 1` for section 1 and section 2.

- `capa2.nsl.msue.edu> cd nsc111f7/TeX`  
`capa2.nsl.msue.edu> latex section1-set1`  
.....  
Output written on `section1-set1.dvi` (3 pages, 16304 bytes).  
Transcript written on `section1-set1.log`.

The above message shows that the `.dvi` file has been created from the `.tex` file for section 1.

- `capa2.nsl.msue.edu> latex section2-set1`  
.....  
Output written on `section2-set1.dvi` (2 pages, 11108 bytes).  
Transcript written on `section2-set1.log`.

Another `.dvi` file has been created, this time for section 2.

- `capa2.nsl.msue.edu> dvips section1-set1.dvi -o section1-set1.ps`  
.....  
This command generates the `.ps` file which is ready for printing.
- `capa2.nsl.msue.edu> dvips section2-set1.dvi -o section2-set1.ps`  
.....

Another `.ps` file has been created, this time for section 2.

- **capa2.nssl.msu.edu> ls**

```
section1-set1.aux  section1-set1.log  section1-set1.tex  section2-set1.dvi  section2-set1.tex
section1-set1.dvi  section1-set1.ps   section2-set1.aux  section2-set1.log  section2-set1.ps
```

The `ls` command lists the files in the current working directory. As can be seen above, there are five files created for each section. The two `.ps` files contain problems sets for all students in sections 1 and section 2. Two sided printing is very useful for sets which fit on 2 pages. Here, set 1 fits on one page and is printed below on the local printer.

- **capa2.nssl.msu.edu> lpr -P Local\_Printer section1-set1.ps**
- **capa2.nssl.msu.edu> lpr -P Local\_Printer section2-set1.ps**

## 5 Specification of CAPA Problem Sets

This Section describes the CAPA grammar and syntax. Examples show how to code problems and create problem sets.

1. **General Description.** The source code for the problems to be solved by students are contained in files labeled `set $x$ .qz` where  $x$  is the problem set number. These files contain all the information needed for each problem as well as the information for the printed hard copy, the computer-screen presentations, and the web display of the problem set. In the simplest case, one might imagine that the file contains just the text to specify the problems for the students. However, each paper is individualized and must contain variable information such as the student's name, the due-date of the set, etc.

CAPA uses the `/DIS()` command to display functions, variables, or anything that has been evaluated and must be inserted in the text when the output is generated. CAPA uses other special commands at the beginning of a line. For example: `/LET` defines a variable. `/IMP` imports a file into the `.qz` file. `/HIN` defines a hint which is available for viewing by the student after the student has entered an incorrect answer. `/ANS()` defines the answer to the problem. Note the UPPER CASE used for the commands. The `//` is used for lines that are comments and should not be evaluated or inserted into the text or displayed to the students. The grammar for writing problem sets with some examples is given in the following section.

2.  **$\LaTeX$  and ASCII in CAPA problems.** A complete but plain problem set can be written using only ASCII characters. In this case the printed version and the version displayed on the VT100 terminal will be exactly the same. However, we have found that high quality printed problem sets can be readily produced by the CAPA system with  $\LaTeX$ . Thus, an important and sometimes confusing aspect of the `set $x$ .qz` files is the  $\LaTeX$  commands to control the presentation of the problem. Greek characters, postscript figures, subscripts and superscripts must be imbedded in the problem set text but cannot be displayed on the VT100 terminal window during interactive logins. Using the `tex()` function it is possible to specify a  $\LaTeX$  version and an ASCII version of a particular string of characters. For example,  $\Delta H$  could appear on paper whereas only `delta-H` can be shown on the screen (using ASCII characters). The general form of the function to be evaluated is `tex("TeX","AScii")`. For example:

```
/DIS(tex("$\Delta H","delta-H"))
```

would be used in the `set $x$ .qz` file to display  $\Delta H$  on the printed paper. The first argument is put into the  $\TeX$  version and the second into the ASCII version. Note that the `/DIS()` command, mentioned above is used to evaluate an expression amid text.

3. **html in CAPA problems.** There are two functions available for specifically entering html code into your problem sets to improve the Web display of the problem set over that of ASCII text. If you do not specify html code, then the ASCII code will be displayed on the web document. The first html specific function is the `web()` function which has similar usage to the `tex()` function. The `web()` function has three fields for entry. They are:

`web("ASCII","TeX","html")`. The second function, `html()`, can include anchors, links, or text to be displayed only on the web document. An example of the use of this function is:

```
/DIS(html("<br><img src=/teacher/nsc111f7/pictures/Ball.gif>"))
```

4. **Grammar Specifications**

- The end-of-line character (produced by the *Return* key) plays a critical role in the `setx.qz` files. It is used to signify the end of text and mathematical expressions. Since single end-of-line characters are ignored by  $\text{\LaTeX}$ , careful usage of the *Return* key can be used to format the ASCII output. Because the `quizzier` display automatically line-wraps the text input, it is important to check the ASCII display before distributing a problem set. The best method is to log in just as a student would.
- It is convenient to begin the question file, `setx.qz` with commands such as `/IMP "HWTtop"` and `/IMP "../Tools/StdMacros"`. These commands ‘import’ a header to display the student’s name, the course name, as well as macros for often used commands and instructions. The questions with their coded answers (and appropriate hints, comments and explanations if desired) follow. `/END` command and a carriage return signify the end of the entire set. It is recommended that each type of problem for a specific field of study be classified and kept in individual ASCII files. This way an entire problem set can be generated by using commands similar to `/IMP "../ProblemLibrary/type04/prob3"`.
- Note that there are also two auxiliary files with commands for formatting the  $\text{\LaTeX}$  2.09 document called `TeXheader` and `Texfooter`. These are automatically included by `quizzier` at the top and bottom of the `quiztemp.tex` file before the `quiztemp.dvi` file is created. The `TeXheader` file used at MSU creates a high density two-column format for the problem sets. This file is contained in the `nsc111f7` directory and is shown below:

```

\documentstyle[twocolumn,epsf]{article}
\textheight 25.9cm
\oddsidemargin = -0.42in
\evensidemargin = -0.42in
\textwidth= 7.28in
\begin{document}
\voffset=-4.1cm
\setcounter{page}{1}
\newcommand{\\capa\}{\s1 C\kern-.10em\raise-.00ex\hbox{\rm A}\kern-.22em%
{\s1 P}\kern-.14em\kern-.01em{\rm A}}}
\newenvironment{choicelist}{\begin{list}}{\setlength{\rightmargin}{0in}
\setlength{\leftmargin}{0.13in}\setlength{\topsep}{0.05in}\setlength{\item
sep}{0.022in}\setlength{\parsep}{0in}\setlength{\belowdisplayskip}{0.04in}
\setlength{\abovedisplayskip}{0.05in}\setlength{\abovedisplayshortskip}
{-0.04in}\setlength{\belowdisplayshortskip}{0.04in}}{\end{list}}

```

- The `HWTtop` file should be at the top of each problem set. It includes the student name, section number, class title, assignment number, due date, and student’s CAPA ID for the assignment.
- The actual `HWTtop` file information used in `nsc111f7` is shown below, together with a magnified example of the resulting output. Note: In the actual set, the carriage returns are omitted from the header information.

```

/LET nameforTeX = name()
/LET due_dateforTeX = due_date()
/LET PINforTeX = capa_id()
/LET sectionforTeX = section()
/LET setforTeX=set()
/LET classforTeX=class()
//
/DIS(tex("\noindent{\large\bf ", ""))/DIS(var_in_tex(nameforTeX))
/DIS(tex("\hfill Section {\Large ", ""))/DIS(var_in_tex(sectionforTeX))
/DIS(tex("}\}\\", "")) /DIS(tex("\vskip -.1in", ""))
/DIS(tex("\noindent {\large \it Sample \capa Questions \hfill Set ", ""))
/DIS(var_in_tex(setforTeX))/DIS(tex("}\}\ \vskip -.16in", ""))
/DIS(tex("\noindent{\bf ", ""))
/DIS(var_in_tex(classforTeX))/DIS(tex("} -- MSU -- Various Dates. Due ", ""))
/DIS(var_in_tex(due_dateforTeX)) /DIS(tex("CAPA ID is", ""))
/DIS(tex("{\footnotesize ", ""))/DIS(var_in_tex(PINforTeX))/DIS(tex("}", ""))
/DIS(tex("\vskip .1in \pagestyle{empty} ", ""))
//
//
/LET DeptID="College of Natural Science, Michigan State University"
/LET stdendlineA=web("",
  "\vskip .05in\hrule width3.55in height .6pt\hfill\vskip -.07in\noindent{\tiny ",
  "<Font color=\\"gray\"><font size =-1><i> --- ")
/LET stdendlineB=web("",
  " \dotfill {CAPA\copyright msu})",
  " --- </font><font size=-2>CAPA &#169</font><font=-1> msu</i></font>
  <font color=\\"black\"")
/LET stdendline=stdendlineA+DeptID+stdendlineB

```

- These lines contain L<sup>A</sup>T<sub>E</sub>X commands for alignment and font specification. They use arguments in the `tex()` function for L<sup>A</sup>T<sub>E</sub>X which are not displayed in the ASCII or Web versions. The functions print the Student Name, section number, problem set number, due date and CAPA ID.
- The L<sup>A</sup>T<sub>E</sub>X source document created from `TeXheader` and `HWTop`, i.e., the first lines in the file `quiztemp.tex` file follow: (again, some carriage returns have been added)

```

\documentstyle[twocolumn,epsf]{article}
\textheight 25.9cm
\oddsidemargin = -0.42in
\evensidemargin = -0.42in
\textwidth= 7.28in
\begin{document}
\voffset=-4.1cm
\setcounter{page}{1}
\newcommand{\capa}{\{\s1 C\kern-.10em\raise-.00ex\hbox{\rm A}\kern-.22em%
  {\s1 P}\kern-.14em\kern-.01em{\rm A}\}}
\newenvironment{choicelist}{\begin{list}{}{\setlength{\rightmargin}{0in}
\setlength{\leftmargin}{0.13in}\setlength{\topsep}{0.05in}
\setlength{\itemsep}{0.022in}
\setlength{\parsep}{0in}\setlength{\belowdisplayskip}{0.04in}
\setlength{\abovedisplayskip}{0.05in}
\setlength{\abovedisplayshortskip}{-0.04in}
\setlength{\belowdisplayshortskip}{0.04in}}{\end{list}}

\noindent{\large\bf Student, Jamie .\hfill Section {\Large 1}}\
\vskip -.1in \noindent
{\large \it Sample \capa Questions \hfill Set 1}\
\vskip -.16in \noindent{\bf nsc111f7}
-- MSU -- Various Dates. Due Fri, May 15, 1998 at 08:00 CAPA ID is
{\footnotesize 5743}
\vskip .1in \pagestyle{empty}

```

Note that all of the CAPA functions have been evaluated before the file is written.

- The result from Latex and dvips processing is a student copy which looks like:

---

<b>Student, Jamie</b>	<b>Section 1</b>
<i>CAPA Natural Science-</i>	<i>Set 1</i>
nsc111f7 – MSU – 1995. Due Sun, Sep 1, 1996 at 08:00. CAPA ID is 8755	

---

- The remainder of the setx.qz file contains the questions. Each question typically consists of a block of variable definitions, the text for the question including evaluation of imbedded data, the coding of the answer, and optionally, a comment, a hint and an explanation.

5. **Example 1.** A complete example of a very simple CAPA problem is given below:

```

/*****
/BEG prob_val=2
/IMP "../Tools/Problem#"
// By S. Wolfe, wolfe@nscl.msu.edu, No commercial use.
/HIN Summing is the same as Addition.
/EXP Add the two numbers and enter the result.
//
/LET firstnumber=random(1,5,1)
/LET secondnumber=random(2,10,2)
//
Find the sum of /DIS(firstnumber) plus /DIS(secondnumber).
//
/LET sum=firstnumber+secondnumber
/ANS(sum,tol=0.01,wgt=prob_val,tries=10)
/DIS(stdline)
/*****

```

The printed output for a specific student appears as:

1. **[2pt]** Find the sum of 3 plus 8.

- 
- Each of the lines in this example is terminated with a carriage return end-of-line character, even though it is not visible here.
  - The `/BEG prob_val=2` begins the problem and sets the weight of the problem to 2 points. (The weight must be an integer from 1 to 9.)
  - The `/IMP "../Tools/Problem#"` function is used to print the problem number and the weight of the problem with a bold font.
  - All lines that start with double-slashes, `//`, are comment lines and only appear in the `setx.qz` file. For example, the comment lines of stars are used to offset problems so that they can be identified easily. As in any computer code, comments play an important role in documentation and “self-documenting” variable names are helpful.
  - The line beginning with `/HIN` contains an optional hint, i.e., it is up to the instructor whether to include such a hint for the students on a particular problem. If a hint has been coded into a problem, the student can ‘request’ to view the hint after attempting to answer the question.
  - The line beginning with `/EXP` is the explanation. The explanation becomes available to the students after data entry to the problem set is ‘closed.’ (Explanations and Hints are included at the discretion of the instructor).
  - The lines beginning with `/LET` are definitions of the variables needed for this problem. `firstnumber`, is declared to be a random integer in the range of 1 to 5 in steps of 1, similarly, `secondnumber` is an integer in the range of 2 to 10 in steps of 2. CAPA displays real and integer variables differently. It is preferable not to mix integer and real numbers. In functions such as `random()` they must not be mixed.
  - Next, we have the text of the problem that will be contained in the printed problem set and shown on the computer screen. Note that the `/DIS()` command around a previously declared variable tells the computer to display its value. If the variable is not defined prior to displaying, a window will display an error message.

- The `/LET sum = . .` declares that the variable `sum` is the sum of the two random numbers. This is the answer to the problem and we use an appropriate variable name. (The name of the answer, here `sum`, is up to the author.)
- Finally, the answer to the problem is declared using the `/ANS()` command. The line with the answer should come literally at the end of all the material for a problem. See Section 6 for explanations of the arguments of the `/ANS()` function.

## 6. Example 2. Using the `choose()` function

The `choose()` function can be used to select among choices or to correlate variables with pictures or graphs. (Note: there should be NO carriage return in the `choose` line. In general, try to minimize carriage returns as they use up the limited amount of lines available as screen display on the vt100 terminals). In this example the computer selects a random number in the range of 1 to 14 and then constructs a filename to be displayed and the correlated answer.

```

/*****
/BEG prob_val=1
/IMP "../Tools/Problem#"
//By E. Kashy, kashy@nscl.msu.edu, No commercial use
/LET k=random(1,14,1)
/LET file=choose(k,"35","45","56","62","77","86","93","m26","m39","m47","m51","m66","m74","m84")
/LET ans5=choose(k, 3.5,4.5,5.6,6.2,7.7,8.6,9.3,-2.6,-3.9,-4.7,-5.1,-6.6,-7.4,-8.4)
The graph shows the function /DIS(tex("\ \centerline{$Y=a x$} ", "Y=ax. "))
Make a careful determination of the value of /DIS(tex("$a$", "a")).
/DIS(tex("\epsfxsize=3.6in \epsffile{/usr/users/teacher/CAPA45/nsc111f7/pictures/Y", ""))
/DIS(var_in_tex(file))/DIS(tex("x.ps"), "")
/DIS( html("<IMG SRC=/nsc111f7/pictures/Y") )/DIS(html(file))/DIS(html("x.gif >") )
/ANS(ans5:2E,tol=5%,wgt=prob_val,tries=10)
/DIS(stdline)
/*****

```

The output on paper would look like (see next page):

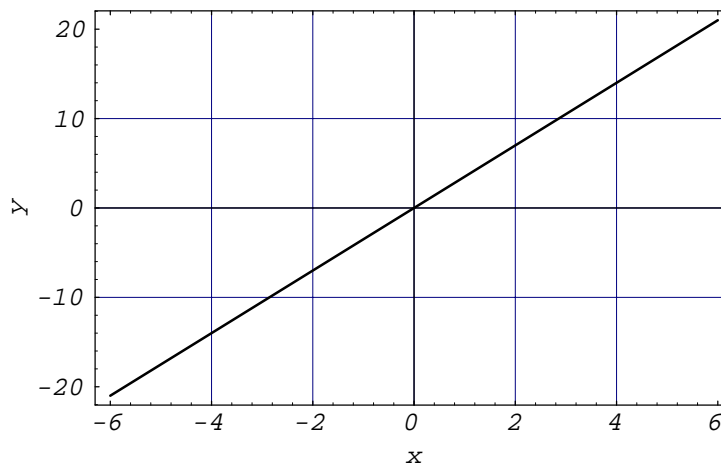


---

2.[1pt] The graph below represents the function

$$Y = ax$$

Make a careful determination of the value of  $a$



- 
- The computer code picks a random number,  $k$ , and the `choose` function selects the string variable, “35”, that will be included in the name of the `.eps` file to be displayed. The `choose` function contains the selection index as its first argument and the choices as the remaining arguments.
  - It is important to check that the number of choices corresponds exactly to the maximum number of the selection index. Otherwise an error will occur whenever the index (randomly) goes outside the range of choices. An ‘Out of bounds’ error message is displayed.
  - The answer is also specified with a `choose()` function using the same selection index.
  - The author of the problem must be sure that the data in the first `choose()` function corresponds to the answers in the second `choose()` function.

7. **Example 3. Scrambling order of N-correct out of M-choices.** The traditional multiple choice problem usually has only one correct answer described by a letter or group of letters. To prevent guessing and to encourage the students to discuss the concepts presented by such problems, we have created problem prototypes where the order and number of choices vary.

In this example a set of variables is loaded for each student and displayed in the text of the problem. The students are then given three statements which are generated from up to four versions.

---

```
//*****
/BEG prob_val=1
/IMP "../Tools/Problem#"
//By E. Kashy, kashy@nscl.msu.edu, No Commercial Use
// ../CAPA4/MCTools/Nof3 Select N correct of 3 Statements
A mass M = 0.16 /DIS(kg_u) is on a smooth horizontal surface (negligible
friction).
It is attached to a spring with spring constant k = 15.3 /DIS(NPm_u). The mass
is
displaced from its equilibrium position (x=0) by a distance of 0.14 m
```

```

(x= + 0.14 m) and then released from rest.
// The next 3 lines are the answer format instructions for the students.
(Give all of the correct answers, i.e., A or BC or ABC...)
//-----
/LET s1a="The motion of M is simple harmonic motion."
/LET s1b="Statement 1 variation b"
/LET s1c="Statement 1 variation c"
/LET s1d="Statement 1 variation d"
/LET mix1=random(1,1,1)
/LET a1a=1
/LET a1b=26
/LET a1c=26
/LET a1d=26
//
/LET s2a="The amplitude varies with time."
/LET s2b="The amplitude doesn't vary with time"
/LET s2c="Statement 2 variation c"
/LET s2d="Statement 2 variation d"
/LET mix2=random(1,2,1)
/LET a2a=2
/LET a2b=1
/LET a2c=26
/LET a2d=26
//
/LET s3a="The frequency of oscillation is independent of k."
/LET s3b="The Period of oscillation is independent of k."
/LET s3c="The frequency of oscillation depends of k."
/LET s3d="Statement 3 variation d"
/LET mix3=random(1,3,1)
/LET a3a=2
/LET a3b=2
/LET a3c=1
/LET a3d=26
//
/IMP "../MCTools/Nof3aux"
/ANS(Nof3right,wgt=prob_val, str=mc, tries=5)
/DIS(stdline)
//*****

```

The output of the above code appears as:

---

1. [1pt] A mass  $M = 0.16 \text{ kg}$  is on a smooth horizontal surface (negligible friction). It is attached to a spring with spring constant  $k = 15.3 \text{ N/m}$ . The mass is displaced from its equilibrium position ( $x=0$ ) by a distance of  $0.14 \text{ m}$  ( $x=+ 0.14 \text{ m}$ ) and then released from rest. (Give all of the correct answers, i.e., A or BC or ABC...)

- A) The amplitude varies with time.
- B) The Period of oscillation is independent of  $k$ .
- C) The motion of  $M$  is simple harmonic motion.

---

The statements displayed to the student are controlled by the `mix $x$ =random(1,4,1)` index. The upper limit on the random function is determined by the number of statements available as options in each set. In the example there is only one choice of the first statement, and `mix1=random(1,1,1)`. For the third statement, there are three choices and `mix3=random(1,3,1)`, so students get one of the other.

The example relies on an auxiliary file to do the scrambling. This file is imported into `set $x$ .qz` with the command:

```
/IMP "../MCTools/Nof3aux"
```

The imported auxillary file depends upon consistency in variable names. Do not change the names of the statement variables (e.g., `s1a`) or the names of the answer variables (e.g., `a1a`). If you wish to make major changes in tool formats, it is recommended that you include the auxillary file within the `setx.qz` rather than using the `import (/IMP)` command. The choices are loaded into specific statement variable names (i.e., `/LET s1a="The motion of M is simple harmonic motion"`) that are correlated to the answer variables for ‘correct’ `/LET a1a=1`; or ‘incorrect’ `/LET a1a=2`. In this example, the variable `s1b` is not used in the problem, so its answer `a1b` is not changed from the value 26. In all the preformatted MCTools files, we have used 26 to indicate an unused answer/statement.

- The file `Nof3` can be simply copied from the `CAPA4/MCTools` directory and pasted into the problem set file. The `Include Button` in `quizzer` allows a user to scan the directory structure and perform that function.
- The various statements (and their variations) are typed into the `s1a`, `s2a`, etc. and corresponding values for `a1a`, `a2a`, etc. indicating if the variation is correct, must be entered.
- The text of the new problem should be typed in.
- Certain variables

```
kg_u, Npm_u, stdline
```

are not defined within the problem. Because they are used often, they were defined in files imported once at the start of the set (`/IMP "../Tools/StdUnits"`) to simplify problem writing.

- The middle integer in `mix3=random(1,3,1)` corresponds to the number of variation of statement block 3.

8. **Using Logical Functions and other Functions.** A number of logical functions are provided in *CAPA* to allow numerical testing for various conditions. The logical expression is tested and returns 0 if False and 1 if true. These expressions can be used in a number of ways, for example in the `rank4aux` file used in problem 11 on page 51

---

## 6 CAPA Functions

1. **Question source text:** CAPA assumes that any characters in the setx.qz without a special command at the beginning of a line will be displayed as text for the student. Those special commands are /LET, /BEG, /IMP, //, /HIN, /EXP, /MAP, /ANS). The information on these special lines can be interleaved with text, but variables must be defined in the text before they are used. For clarity and readability, a regular format with definitions followed by text, followed by the solution is encouraged.

### 2. Variable Definitions and Expressions (/LET )

- Variable definitions and expressions begin on a single line with a /LET command that is terminated by a single end-of-line character. Such entries will be line-wrapped by the editor and may appear to extend past one line on the display.
- The CAPA system does not reserve variable names **except for certain functions used in the system**. They are listed in the Intrinsic Functions Table, and must not be used as variable names.
- Definitions are specified in the form:

/LET VariableName=expression

- **List of Available Expressions:**

<i>Integer</i>	◊ An integer.
<i>Real number</i>	◊ A real number. It could be of the form 123.4, 1.234E+2, 1.234E+2, 1.234E+02, 1.234e+02.
<i>"string"</i>	◊ A string. It is specified in the form "A block of text".
<i>Variable Name</i>	◊ A variable previously defined.
<i>( expression )</i>	◊ Precedence, evaluate <i>expression</i> first.
<i>- expression</i>	◊ Negative of <i>expression</i> .
<i>function([expression, ...])</i>	◊ Call a function with arguments. (see below)
<i>expression * expression</i>	◊ Multiply expressions.
<i>expression / expression</i>	◊ Divide expressions.
<i>expression + expression</i>	◊ Add expressions, concatenate strings
<i>expression - expression</i>	◊ Subtract expressions.
<i>expression == expression</i>	◊ Logical; expressions equal? Returns 0 if false, 1 if true
<i>expression != expression</i>	◊ Logical; expressions different? Returns 0 or 1
<i>expression &gt;= expression</i>	◊ Logical; greater than or equal? Returns 0 or 1
<i>expression &lt;= expression</i>	◊ Logical; less than or equal? Returns 0 or 1
<i>expression &gt; expression</i>	◊ Logical; greater than? Returns 0 or 1
<i>expression &lt; expression</i>	◊ Logical; smaller than? Returns 0 or 1

- The variable names must begin with a letter but may contain letters, numbers, and underline characters.
- Variable names (and function names) are case sensitive, and there is no limit on the length of a variable name.
- There are three types of variables, integer, real and string. The type is not explicitly specified but rather is assigned by context at the time the variable is defined.

- Variables must be defined before they are used in any other expression. Be careful to avoid mixed integer and real expressions, this is particularly important in functions such as “random” and “pow(er)”. Examples of how a variable may be defined:

```
/LET PI=3.1415926535897
/LET two_pi=2.*PI
/LET r1=random(1,2,1)
/LET string1=choose(r1, "Hello", "Goodbye")
```

- **Note:** Quotation marks indicate the beginning and end of strings. You must use a backslash to display the quotation marks within a string. For example: `/LET string="I'll be back."` will produce the output "I'll be back."
- Do not use intrinsic function names (ie; sin, pow, etc. ) for variable names.
- Once defined, the same variable can be used at any subsequent point in the entire problem set and it can even be redefined. Therefore, variable names that are unique to specific problems help to limit their scope.
- The *value* of a previously defined variable can be displayed in text by placing the variable name in the `/DIS()` command, e.g., `/DIS(r1)`. The format of the display value can be controlled as well using a colon and a specification. For example, `/DIS(PI:3f)` means display the variable 'PI' as a floating point number with three places after the decimal. The colon is a delimiter indicating that a format specification follows. `/DIS(var:2E)` will display the variable with two decimal places in scientific notation.
- The very end of the text and definitions, hints, explanation and solution for a given problem is indicated by the position of the `/ANS()` command.
- Lines longer than 80 characters are wrapped around to the next screen line at the nearest break between words on the screen by the **quizzer** editor.
- **Note:** A total of 37 lines are available for display on the VT100 terminal. This is divided into two pages with 20 lines available on the first page and 17 available on the second with three lines repeated for reading consistency. Hints and explanations are displayed on separate screens, each of which can have 20 lines. There are no limits on the printed or Web versions.

### 3. Variable Definitions and Expressions (`/BEG` )

- `/BEG` is simply an alias for `/LET`, and is used to begin each problem in the following manner: `/BEG prob_val=3`. This sets the variable `prob_val` to 3. `prob_val` can then be used to define the weight of the problem in the `/ANS` specification. Weights are integers and can range from 0 to 9 points.

### 4. The Import function (`/IMP`)

- The `/IMP "filename"` function calls and uses the contents of the specified file, however the content is not displayed in the source file.
- The `/IMP` command requires a string input, either the string filename or a variable which provides the string filename:

```
/IMP "HWTop"
```

will import the HWTop file from the local directory (for files in other directories, proper UNIX paths must be included as part of the filename). e.g., `/IMP "../Tools/StdMacros"`; The file may be selected dynamically from a set, for example,

```
/LET integer=random(1,10,1)
/LET filename="File"+integer
/IMP filename
```

which will import a randomly selected file $X$  from File1, File2, ...File10. ( Note that adding an **integer** to a **string** results in a concatenated **newstring**.) This can be used to provide even greater variability among sets for students, since different problems on a given topic can be selected. This feature may be especially useful to produce standardized tests. With a large enough problem base, students can be given a randomly selected set to prepare for the test.

## 5. Comment Lines (//)

- Comments are character strings on lines that start with a double slash, //, and contain no carriage returns.
- Comments can be written on the same line as other statements. For example:  
/LET number=random(1,4,1) // chooses a random number
- The comments are only displayed by the **quizzer** module and are for the benefit of the people writing and reviewing the actual problem code.
- Instructors are strongly encouraged to include comment lines in order to describe the structure of the problems for future use.

## 6. Hints (/HIN)

- Hints are optional. There can be one hint per problem. When coded into a specific problem the hint is made available to the students after an answer (correct or incorrect) has been proffered.
- The text for hints contain character strings (no expressions) and can display string variables defined before the hint to match a problems content, and is specified as:

```
/LET index=random(1,4)
/LET pronoun=choose(index,"he","she","she","he")
/LET ppronoun=choose(index,"his","hers","hers","his")
/LET person=choose(index,"son","daughter","niece","nephew")
/HIN The /DIS(person) was being pulled up before /DIS(pronoun)
/HIN let /DIS(ppronoun) balloon escape.
```

- This hint has two lines and two carriage returns. The contents of both lines are displayed simultaneously; it is not two separate hints.
- Only one /HIN is needed if there is only one carriage return at the end of a hint consisting of several lines of text.
- Content of all lines typed before the answer with a /HIN at the beginning are displayed as the hint. The relative ordering of the hint lines is preserved, and the hint is displayed as a separate page on the VT100 screen and below the answer box on the web version.
- Hints can also be viewed after a problem set is *closed*.

## 7. Explanations (/EXP)

- Lines of explanations begin with the /EXP characters and are displayed on the login-terminal only when requested after a problem set is closed.
- Explanations to a given problem are very similar to hints. It is sometimes useful to also include a detailed explanation for problems, in contrast to the hints. Students often review old problem sets at exam time and may not exactly remember how to solve a specific problem. Explanations are also optional.

```
/EXP A good explanation should remind the student
/EXP how to do the problem, and might take several lines.
```

## 8. The mapping function (/MAP)

- The /MAP function is used to map and permute one set of variables onto another.
- For example:

```
/LET seed=random(1,300,1)
/MAP(seed;M1,M2,M3;m,n,o)
```

Assigns to the variables M1, M2, and M3 the values of the variables m, n, and o. The correspondence (i.e., which of m, n or o is assigned to M1, etc....) is determined by the value of the variable 'seed', which in this example is selected from 1 to 300.

- The /MAP function is used in all the auxiliary files of the multiple choice templates prepared to facilitate coding of qualitative or conceptual questions.

## 9. Answers (/ANS())

- Individual problems are ended by an expression specifying the answer. This answer expression consists of the keyword /ANS starting from the first character of a line.
- The answers to problems are previously defined variables. The answers are indicated specifically by the /ANS() command, with the parentheses () containing the attributes of the answer. The attributes of the /ANS() function are **case insensitive**. The minimum attributes of a basic answer statement are the **variable** defining the answer, the **problem weight** and the number of **tries**. The possible attributes to the /ANS() command are:
  - (a) **Tolerance:** Tolerance can be specified as a set numerical value, as a percentage of the correct answer (after it has been formatted) or as a variable. The format for the set numerical value would be given by: /ANS(variable, tol=0.5, tries=5, wgt=1) The format of a percentage tolerance would be given by: /ANS(variable, tol=1.1%, tries=5, wgt=1). The format for the tolerance as a variable would be: /ANS(variable, tol=TolVar, tries=5, wgt=1).
  - (b) **Problem Weight:** The problem point value (or weight) is set by: /ANS(variable, wgt=2, tries=5) The problem value can be an integer value between 0 and 9. This can be specified at the beginning of a problem by: /BEG prob\_val=3 and having this value called in the /ANS() function by: /ANS(variable, wgt=prob\_val, tries=5).
  - (c) **Significant Figures:** If unspecified, the maximum number of allowed significant figures is 15. The number of acceptable significant figures can be specified by the following format: /ANS(variable, sig=4 plus 1 minus 1, tries=5, wgt=1) This sample format will accept 3, 4, or 5 significant figures in the answer.
  - (d) **Units:** A file, capa.config in the class directory lists all acceptable units and their relationships, and that file can be edited by the instructor. All SI units are included. A typical format is /ANS(perimeter:3f,tol=0.009,wgt=prob\_val,tries=10,unit="cm",sig=4)) or /ANS(area:2f,tol=1.1% ,wgt=prob\_val,tries=6,unit="cm^2",sig=3 plus 1)) A file that shows the way units are entered is provided for the Web version ../Links/UnitFmt.html. Its content can be provided as a HINT in both the Web or VT100 student options.
  - (e) **Hand Grading:** A problem can be designated as hand graded only. For this case students are not allowed to answer via telnet or web sessions and the question is graded by the instructor. The format for this is: /ANS(variable,HGR=on, tries=1, wgt=1)

- (f) **Partial Credit:** A multiple point problem can be specified to have partial credit. This attribute has the same functionality of `HGR=on` in that students cannot login to answer the question. The format for this is: `/ANS(variable, wgt=9, PCR=on, tries=1)` This allows an instructor to manually enter a point value between 0 and 9 through the `grader` application.
- (g) **Tries:** The number of attempts to enter a correct answer can be limited to an integer value less than 99. The format is: `/ANS(variable, tries=35, wgt=1)` Errors in significant figures or error in units will not decrease the number of remaining tries, but they must be satisfactory before a `Correct` or `Incorrect` is given. The student receives a message near the answer field stating the number of attempts and the total number of tries available for each problem in which the `tries=x` has been specified. A warning message is displayed when only one try remains.
- **Integer answers** can require the exact answer or have an integer as tolerance.
    - (a) `/LET AnsVar=34`  
`/ANS(AnsVar, wgt=1, tries=5)` Only literally the number 34 will be accepted as correct.
    - (b) `/ANS(Sumx, wgt=1, tries=5)` Assuming that Sumx is a previously calculated or defined integer, only its exact value will be accepted as correct.
    - (c) `/ANS(Sumx,tol=4, wgt=1, tries=5)` If Sumx is a previously calculated or defined integer, then any answer within plus or minus 4 units is accepted as correct.
    - (d) `/ANS(Sumx,Tol=4,Wgt=3, tries=5)` Sumx is a previously calculated or defined integer, and any answer within plus or minus 4 units is correct. It is assigned a weight 3 points in calculating the score. Note that the default value of Wgt is 1 point
  - **Real answers** A tolerance must be specified for 'real' answers. It can absolute or relative. An absolute tolerance can be a variable.
 

An answer formatted to have a given number of significant figures can be required. If the answer submitted does not conform, students are sent the message "Adjust Sig.Figs" and must reenter their solution. They are then told if the answer is correct. Answers with real values can be formatted with specifications `':1E'`, `':3E'`, `':2f'`, etc., where `:1E`=exponential notation with one decimal place, `:3E`=exponential notation with three decimal places, and `:2f`=two floating decimals.

    - (a) `/ANS(speed:2E,tol=1.2%,Wgt=prob_val, tries=5)` When an answer submitted by a student is within the relative tolerance (1.2% ) of the correct answer, the student gets `'correct'`. **Note:** The relative tolerances are based on the formatted answer. If the calculated answer is 100.1 and `:2E` formatting is used, the tolerance is based on the value 1.00E02. For example if 101.2 is entered, the message reads `Correct, Computer gets 1.00E02`. If the variable `prob_val` was defined in the `/BEG prob_val=3` command at the beginning of a problem, then the weight of the problem is 3 points
    - (b) `/ANS(speed:2E,tol=35.0,Wgt=2, tries=5)` When an answer submitted by a student is within the absolute tolerance of 35.0, the student would get the problem correct. For example if 1350.0 is entered and is the correct answer, the message reads `Correct, Computer gets 1.35E03`. This absolute tolerance is also based upon the formatted answer. The limit of accepted answers are the formatted answer  $\pm$  the tolerance.
    - (c) `/ANS(speed:3E,tol=qty3,Sig=3 plus 1,Wgt=prob_val, tries=5)` Here the tolerance is a previously defined quantity, `qty3`. The student's answer must have either 3 or 4 significant figures, or else the computer response is `Adjust Sig.Figs`. An acceptable answer will produce the response: `Correct, Computer gets 1.347E03`. `Sig=3` requires 3 significant figures, `Sig=3 plus 1 minus 1` is equivalent to `Sig=2 plus 2`, requiring 2, 3 or 4 significant figures.
    - (d) `/ANS(speed:3E,tol=qty3,Sig=3 plus 1,Wgt=4,Pcr=on,tries=5)` `Pcr=on` allows instructor to enter a value from 0 to 4 for that problem, which could, for example, be an essay or derivation or a diagram, etc..



- **String answers** can be exact comparisons, or have the order and/or case disregarded.
  - (a) `/ANS("Kc1",wgt=prob_val,tries=5)` Only the strings `Kc1 kc1 KcL . . .` are correct, as default is case insensitive.
  - (b) `/ANS(i5p5right,wgt=prob_val,tries=5)` `i5p5right` is a string variable previously defined. The order is essential, but not case sensitive , `AdcEb adceb ADCEB` are equivalent.)
  - (c) `/ANS(letters,Str=MC,wgt=prob_val,tries=5)` `letters` is a string variable previously defined. The command `Str=MC` accepts any order of the correct responses. That is, `abc` is equivalent to `acb`, etc..
  - (d) `/ANS(letters,Str=CS,wgt=prob_val,tries=5)` `letters` is a string variable previously defined. The order and the case are required, as it is an exact string comparison.
- 10. `/END /END()` or `/END(stdendline)` prompts the parser to stop parsing and ignore all input characters beyond that command. Material between the last `/ANS()` and `/END` is not included in the output; however, string variables can be defined and constructed. All sets should include `/END(stdendline)`, so that both on paper and on the Web, the department is identified and the copyright nature of the *CAPA* software is indicated. The string for `stdendline` is created in the `HWTop` file in the class directory and can be edited by users.
- 11. The *CAPA* system counts the number of problems in a problem set by counting the number of proper *answer-lines*. If there is an error in coding and the answer can not be evaluated then the “problem” is not counted by `quizzer`.

Table of Intrinsic Functions: names are case sensitive.	
Functions	Description, Sample quizzer input and output for printing
sin(x), cos(x), tan(x)	<p>Trigonometric functions. x is in radians.</p> <pre>/LET angle=60.0 /LET var1=sin(angle*3.141592654/180) Sine of /DIS(angle:1f) degrees is /DIS(var1:3f).</pre> <p>Output:Sine of 60.0 degrees is 0.866.</p>
asin(x), acos(x), atan(x)	<p>Inverse trigonometric functions. Returns radians</p> <pre>/LET PI=3.141592654 /LET var1=-1.65 /LET var2r=atan(var1) /LET var3d=atan(var1)*180.0/PI The angle whose tangent is /DIS(var1:3f) can be expressed as /DIS(var2r:3E) radians or as /DIS(var3d:3E) degrees.</pre> <p>Output:The angle whose tangent is -1.650 can be expressed as <math>-1.026</math> radians or as <math>-5.878 \times 10^1</math> degrees.</p>
log(x), log <sub>10</sub> (x)	<p>Natural logarithm and base-10 logarithm. Note that the the variables natlog and tenlog were defined to simplify the coding of this problem by allowing complicated text (with subscripts and superscripts) to be printed by displaying a variable.</p> <pre>/LET natlog=tex("\$\log_e\$", "log_e") /LET tenlog=tex("\$\log_{10}\$", "log_10") /LET x=2546.7 /LET var1=log(x) /LET var2=log10(x) /LET ratio=var1/var2 /LET var3=log(10.0) The /DIS(natlog) of /DIS(x:1f) is /DIS(var1:3E) while its /DIS(tenlog) is /DIS(var2:3E). Note that the ratio /DIS(natlog)//DIS(tenlog) = /DIS(ratio:3f), which is just /DIS(natlog)(10), ie, /DIS(var3:3E).</pre> <p>Output:The <math>\log_e</math> of 2546.7 is 7.843 while its <math>\log_{10}</math> is 3.406. Note that the ratio <math>\log_e/\log_{10} = 2.303</math>, which is just <math>\log_e(10)</math>, ie, 2.303.</p>
exp(x), pow(x,y), sqrt(x)	<p>Exponential, power, and square root. Compute <math>e^x</math>, <math>x^y</math>, and <math>\sqrt{x}</math> respectively.</p> <pre>/LET varx=1.526 /LET vary=0.5 /LET var1=exp(varx) /LET var2=pow(varx,0.5) /LET var3=pow(varx,vary) /LET var4=sqrt(varx) var1=/DIS(var1:2f); var2=/DIS(var2:2f); var3=/DIS(var3:2f); var4=/DIS(var4:2f).</pre> <p>Output:var1 = 4.60; var2 = 1.24; var3 = 1.24; var4 = 1.24.</p>

Functions	Description, sample quizzer code and output
<code>abs(x)</code> , <code>sgn(x)</code>	<p><code>abs(x)</code> returns the absolute value of <math>x</math>. <code>sgn(x)</code> returns 1, 0 or <math>-1</math> depending on value of <math>x</math></p> <pre> /LET xx=-2.5 /LET var1=abs(-4.5) /LET var2=sgn(xx) /LET var3=sgn(-4.5*xx) var1=/DIS(var1:2f) var2=/DIS(var2) var2=/DIS(var2:2f) var3=/DIS(var3) </pre> <p>Output:var1=4.50 var2=-1 var2=-1.00 var3=1</p>
<code>erf(x)</code> , <code>erfc(x)</code>	<p>Error functions. <math>erf(x) = \frac{2}{\sqrt{\pi}} \int_0^x e^{-t^2} dt</math> and <math>erfc(x) = 1.0 - erf(x)</math>.</p> <pre> /LET varx=0.51 Evaluate the Normal Probability Integral from -/DIS(varx) to /DIS(varx). /DIS(newline) /LET prob = erf(varx/sqrt(2.0)) Probability = /DIS(prob:4f) </pre> <p>Output:Evaluate the Normal Probability Integral from -0.51 to 0.51. Probability = 0.3899</p>
<code>ceil(x)</code> , <code>floor(x)</code>	<p>Ceiling and floor functions. <math>ceil(x) = \lceil x \rceil</math> and <math>floor(x) = \lfloor x \rfloor</math>. The ceiling function returns an integer rounding <math>x</math> toward positive infinity. The floor function rounds toward negative infinity.</p> <pre> /LET varx=ceil(3.65) /LET vary=floor(3.65) /LET varz=ceil(-5.73) /LET varw=floor(-5.73) varx=/DIS(varx) vary=/DIS(vary) varz=/DIS(varz) varw=/DIS(varw) </pre> <p>Output:varx=4 vary=3 varz=-5 varw=-6</p>
<code>min(...)</code> , <code>max(...)</code>	<p>Minimum and maximum functions, with indefinite number of arguments. Arguments must be of either all integer or real type. String comparison are such that <math>A &lt; a</math> i.e., lower case letters have a higher value than upper case letters.</p> <pre> /LET a=min(23,45,12,7,9) /LET b=max(23.1,45.3,12.6,7.1,9.0) /LET c=min("a","ae","aeg","Aeg") /LET d=max("a","ae","aeg","Aeg") a=/DIS(a:2f) b=/DIS(b:2f) c=/DIS(c) d=/DIS(d) </pre> <p>Output:a=7 b=45.30 c=Aeg d=aeg</p>

Functions	Description, sample quizzer code and output
factorial(n)	<p>Arguments must be an integer. Returns an integer if n is or smaller, else or real value.</p> <pre> /LET number=factorial(6) /DIS("6!=")/DIS(number)/DIS(newline) /DIS("0!=")/DIS(factorial(0))/DIS(newline) /LET number2=factorial(21) /DIS("21!=")/DIS(number2:3E)/DIS(newline) </pre> <p>Output:6!=720 0!=1 21!=5.109 × 10<sup>19</sup></p>
N% M	<p>N and M are integers, and the remainder of the integer ratio is returned.</p> <pre> /LET ratio = 98/5 /DIS("ratio=")/DIS(ratio)/DIS(newline) /LET leftover=98%5 /DIS("remainder=")/DIS(leftover) </pre> <p>Output:ratio=19 remainder=3</p>
sinh(x), cosh(x), tanh(x)	Hyperbolic functions.
asinh(x), acosh(x), atanh(x)	Inverse hyperbolic functions.
roundto(var1,n)	<p>Rounds the value of real variable 'var1' to n places.</p> <pre> /LET value=100.0/3.0 Initial value = /DIS(value:5f)/DIS(newline) /LET value=roundto(value,2) The new value is now /DIS(value:5f) and can be used as /DIS(value:2f) with no rounding error. </pre> <p>Output Initial value = 33.33333 The new value is now 33.33000 and can be used as 33.33 with no rounding error.</p>
web("a","b","c") or web(a,b,c)	<p>ASCII (a), tex(b) and html(c) strings or variables displayed respectively. Example: {/DIS(web("M2","M\$_2\$","M&lt;sub&gt;2&lt;/sub&gt;")) }</p>
html(a) or html("a")	<p>variable or string 'a' parsed for WWW Browser Only. Example: /DIS(html("&lt;br&gt;&lt;a href=/nsc111f7/Links/webPulley.html&gt; Motion of Masses on a Pulley&lt;/a&gt;&lt;p&gt;"))</p>
asinh(x), acosh(x), atanh(x)	Inverse hyperbolic functions.

Functions	Description, sample quizzer code and output
<code>jn(0,x)</code> , <code>jn(1,x)</code> , <code>jn(n,x)</code>	Bessel functions of the first kind, with orders 0, 1 and $n$ respectively. Note that the first argument is integer, the second real.  <pre> /LET aa=jn(0,3.0) /LET bb=jn(1,3.0) /LET cc=jn(2,3.0) aa=/DIS(aa:4f) bb=/DIS(bb:4f) </pre> Output:aa=-0.2601 bb=0.3391 cc=0.4861
<code>yn(0,x)</code> , <code>yn(1,x)</code> , <code>yn(n,x)</code>	Bessel functions of the second kind, with orders 0, 1 and $n$ respectively. Note that the first argument is integer, the second real.  <pre> /LET dd=yn(1,3.0) dd=/DIS(dd:4f) </pre> Output:dd=0.3247
<code>random(l,u,d)</code> , <code>random(l,u)</code>	Returns a uniformly distributed random number between $l$ and $u$ with steps of $d$ . If not specified $d = 1$ . Note that all arguments must be the same type, integer or real.  <pre> /LET index=random(2,5,2) //returns integers 2 or 4 /LET value=random(2.3,5.15,0.2) //returns 2.3, 2.5, 2.7, ..., 5.1 index=/DIS(index) value=/DIS(value:2f) </pre> Output:index=4 value=3.70
<code>choose(i,...)</code>	Choose the $i$ th item in the argument list. Integer $i$ must greater than zero and it's maximum possible value must not exceed the number of arguments following it.  <pre> /LET indx=3 /LET realvar=choose(indx, 23.0,45.4,67.3) // will select 67.3 /LET intvar=choose(indx, 23,45,67) // will select 67 /LET stringvar=choose(indx,"No","Yes","Maybe") // will select "Maybe" realvar=/DIS(realvar:2f) intvar=/DIS(intvar) stringvar=/DIS(stringvar) </pre> Output:realvar=67.30 intvar=67 stringvar=Maybe
<code>tex(a,b)</code> , <code>tex("a","b")</code>	When in TeX mode, return the first argument $a$ , and return the second argument $b$ in enscript mode (ASCII).  <pre> /DIS(tex("This, in the .tex file.,"This, in the ASCII version.")) /LET A=33 /LET B=66 /DIS(tex(A,B)) </pre> Output:This, in the .tex file. 33
<code>var_in_tex(a)</code>	Equivalent to <code>tex("a","")</code>

Functions	Description, sample quizzer code and output
to_string(x), to_string(x,y)	<p>If variable x is an integer, to_string(x) returns a string. If x is real the format is given by y as follows:</p> <pre> /LET name1=to_string(345) /LET name2=to_string(34.5,".2E") /LET name3=to_string(34.5,".2f") /LET sentence1= "This number is " + name1 + "." /LET sentence2= "This number is " + name2 + "." /LET sentence3= "This number is " + name3 + "." /DIS(sentence1) /DIS(sentence2) /DIS(sentence3) </pre> <p>Output:This number is 345. This number is 3.45E+01. This number is 34.50.</p>
capa_id(), class(), section(), set(), problem()	<p>The CAPA ID number, class name, section number, set number and problem number respectively. Variables can be assigned their current values or they can be displayed directly as shown below.</p> <pre> /LET capaIDval=capa_id() Your CAPA ID is /DIS(capaIDval). /DIS(newline) The class name entered at login is /DIS(class()). /DIS(newline) Your section number is /DIS(section()). /DIS(newline) This is part of problem set /DIS(set()). /DIS(newline) The current problem number is /DIS(problem()). /DIS(newline) </pre> <p>Output:Your CAPA ID is 8659. The class name entered at login is nsc111f7. Your section number is 1. This is part of problem set 2. The current problem number is 2.</p>
name(), student_number()	<p>student name and student number; The latter is can be printed on quizzes when capa_id is used for identification to resolve ambiguities.</p> <pre> /DIS(name()) /DIS(student_number()) </pre> <p>Output:Student, Jamie A12345678</p>
open_date(), due_date(), answer_date()	<p>Problem set open date, due date, and answer date.</p> <pre> /DIS(open_date()) /DIS(newline) /DIS(due_date()) /DIS(newline) /DIS(answer_date()) /DIS(newline) /DIS(due_day()) /DIS(newline) </pre> <p>Output:Tue, Dec 3, 1996 at 08:00. Wed, Dec 11, 1996 at 08:00. Thr, Dec 12, 1996 at 08:00. Wed, Dec 11, 1996.</p>

---

## 7 Auxiliary Files

---

Template files that facilitate problem-coding and improve the legibility of the set source file have been written. They have proven to be very useful for multiple choice conceptual problem-coding and can be expanded and modified to suit a user's particular needs.

1. `/teacher/CAPA45/Tools` directory holds the following files:

```
AnonymousSN Problem# StdMacros StdUnits
parProblem#
```

They include some variable definitions that can be used to simplify problem-coding.

2. `/teacher/CAPA45/MCTools` directory contains files that were written to simplify the coding of qualitative conceptual questions. Answers types include T/F, matching, ranking, etc.. The code in `exam1of8` transforms a single numerical answer into a multiple choice format with 8 choices. `Set2.qz` of the directory `nsc111f7` contains all these problem types in `.qz` format. The directory currently contains the following files:

```
90.120grids Rank5 fig13to13w i3p6 label11auxw
90.60grids Rank5aux fig4to4auxw i3p6aux label11w
90.90grids SM3 fig4to4w i3p6auxw label12auxw
M3T7 SM3aux fig5to5 i4p5 label12w
M3T7aux SM4 fig5to5auxw i4p5aux label13auxw
M4T7 SM4aux fig5to5auxw i4p5auxw label13w
M4T7aux SM5 fig5to5w i4p6 label14auxw
M5T7 SM5aux fig6to6 i4p6aux label14w
M5T7aux SM6 fig6to6auxw i4p6auxw label15auxw
M6T7 SM6aux fig6to6auxw i5p5 label15w
M6T7aux SM7 fig6to6w i5p5aux label16auxw
M7T7 SM7aux fig7to7 i5p5auxb label16w
M7T7aux SM8 fig7to7auxw i5p5auxw label17auxw
M8T7 SM8aux fig7to7auxw i5p6 label17w
M8T7aux SM9 fig7to7w i5p6aux label18auxw
M9T7 SM9aux fig8to8 i5p6auxw label18w
M9T7aux aReadMe_Nof3 fig8to8auxw i6p5 label2auxw
Nof3 aReadMe_ixpy fig8to8auxw i6p5aux label2w
Nof3aux aTypeSummary fig8to8w i6p5auxw label3auxw
Nof4 exam0f_aux fig9to9 i6p6 label3w
Nof4aux exam1f_aux fig9to9auxw i6p6aux label4auxw
Nof5 exam1of8 fig9to9auxw i6p6auxw label4w
Nof5aux exam1of8auxw fig9to9w i7p5 label5auxw
Nof6 exam2e_aux i13p13auxw i7p5aux label5w
Nof6aux exam2f_aux i2p5 i7p5auxw label6auxw
Nof7 exam3f_aux i2p5auxw i7p6 label6w
Nof7aux fig10to10auxw i2p5auxw i7p6auxw label7auxw
Nof8 fig10to10w i2p6 i7p6auxw label7w
Nof8aux fig11to11auxw i2p6auxw i9p9 label8auxw
```

Nof9	fig11to11w	i2p6auxw	i9p9aux	label8w
Nof9aux	fig12to12auxw	i3p5	i9p9auxw	label9auxw
Rank4	fig12to12w	i3p5aux	label10auxw	label9w
Rank4aux	fig13to13auxw	i3p5auxw	label10w	rank3aux

3. `/teacher/CAPA45/nsc111f7/HWTop`: This file contains the header for assignments or exams, as well as defining `stdendline` (standard end line). The file must be edited to reflect the name of the course, department, and University. Note that it is coded so that its information is not part of the text sent to vt100 terminals as this would waste several lines on the screen each time the first problem is displayed.



---

## **8 Sample Login Instructions**

---

An example of login instructions for students is given on the following page.

Department of Physics and Astronomy M. S. U.  
phy183f7

### Instructions for Using CAPA<sup>1</sup>

CAPA implements a **Computer-Assisted Personalized Approach** for assignments, quizzes and examination. Using CAPA in this class is **OPTIONAL**.

Your **Name** and a specific **CAPA ID** is on each assignment has . Solve the problems. Discussion of concepts and methods with fellow students is encouraged. It is strongly recommended that you answer the questions and work the problems before you 'login' to the computer. As you will see when you login, for some problems a :H (for Hint) shows up as an option on the screen once you have tried a problem.

#### I. Access via vt100 terminal emulation.

1. You should have no difficulty finding a vt100 terminal emulator to use. There are numerous computers labs on Campus. Some of these are listed below with the approximate number of stations.

Labs	PCs	MACs	Labs	PCs	MACs
216 BH	33		210 CC		16
222 SKH	33		313 CC	17	
B38 MCD	50		403 CC		17
15 WON	66		415 CC	17	
128 HUB	13	16	500 CC	16	
219 NKL		32	501 CC	17	
112 OHP		28	WIL	40	14
339 CSE	25		Union	40	25
120 CEM	29	23	120 CC	17	

2. Four ways to start a telnet session:

**A.** On most campus PC's, First select the **Class Software** option from the main menu. Choose **CAPA - Assignments**, then choose **Login to capa2**. [Follow the instructions. At the **password:** prompt, press the return key.] You should next see the prompt **login:**

**B.** For most the campus MACs, open the **Class Software** folder and then open the **CAPA** folder. Then open the folder named **capa2**. You should next see the prompt **login:**

**C.** From wherever you can do a **telnet**, enter **telnet capa2.nsl.msue.edu**  
You should next see the prompt **login:**

**D.** Out-of-town access via modem and a local call using the MichNet network is also possible. Check with the Prof.

3. At the '**login:**' prompt, enter **phy183f7** (note the small letters in phy183f7). Your terminal screen should then look like:

```

CAPA, a Computer-Assisted Personalized Approach
-- Course Name and Number --

Enter STUDENT NUMBER and CAPA_ID (hit ENTER/RETURN after each)

STUDENT NUMBER: 
                  ^
                  |
                  | Correct cursor position
CAPA ID: 

To exit system, just hit ENTER/RETURN

```

The cursor on your screen must be in the position shown above. If it is not, your terminal is NOT set as a **vt100**.

4. Enter your **Student Number**, then the **CAPA ID** which is at the top of your assignment.
5. Follow the instructions.

#### II. Access via www browser.

1. Point your browser to URL:  
**http://capa2.nsl.msue.edu/class.html**
2. Choose **phy183f7** from the pop-up menu.
3. Enter your Student Number and CAPA ID which is at the top of your assignment.
4. Click on the appropriate button to try your set.

**Note 1.** You may repeat the problems you missed. Your instructor sets the number of tries. If you get an 'incorrect' response, there is help available for some of the problems. You may login/logout as many times as you wish.

**Note 2.** Do not 'open' multiple sessions or browsers. Be sure to eXit properly each time to terminate a session.

**Note 3.** Avoid using the "Reload/Refresh" button when using the WWW.

Clicking that button after submitting an answer which is incorrect will re-submit the incorrect answer and use up one Try.

<sup>1</sup>CAPA ©1993 - 1997, MSU Board of Trustees.

---

## 9 A Problem Set with Source Code for some Questions

---

An example of a set and *CAPA* source is given on the following pages. Two-sided (duplex) laser printing has been found to be very useful when making assignments or exams that require more than one page.

In physics and chemistry, several users have agreed to share and exchange problems for use in their classes. In physics, this is evolving into the exchange of local physics libraries, i.e., MsuPhysLib, ..., among users. Authors retain commercial rights. A comment line within each problem identifies the author. An e-mail address is useful if some question about the problem or its coded solution arises. Users are also free to edit and adapt the problem for their classes.

Student, Jamie .

Section 1

Sample CAPA Questions

Set 1

nsc111f7 – MSU – Various Dates. Due Fri, May 15, 1998  
at 08:00 CAPA ID is 5743

1. [1pt] The following are possible ways to express the quantity 0.492 (Give ALL correct answers, i.e., B, AC, BCD...) Note: 3.45E-8 is a way you can enter the number  $3.45 \times 10^{-8}$  in most computers.

- A) 4.92E-1
- B)  $49.2 \times 10^{-2}$
- C) 0.492E+1
- D) 0.000492E+3
- E)  $0.00492 \times 10^3$
- F) 0.0492E+1

2. [1pt] Calculate the perimeter of a rectangle with a length of 21.5 cm and a width of 48 mm. Enter units.

3. [1pt] Calculate, in cm, the perimeter of a rectangle with a length of 18.5 cm and a width of 44 mm.

4. [1pt] Calculate the perimeter of a rectangle with a length of 22.50 cm and a width of 44.0 mm.

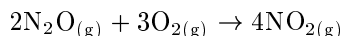
5. [1pt] The lower portion of a flag pole has a height of 2.245 m and is red. Above it the pole is white for a height of 167.2 cm. The top section is blue and has a height of 1756 mm. Calculate the total height of the pole.

6. [1pt] Calculate the area of a rectangle with a length of 13.0 cm and a width of 100 mm.

7. [1pt] Use your calculator to find the square root of  $5.377 \times 10^{-27}$ .

8. [1pt] Calculate the volume of a spherical balloon which has a surface area of  $0.0597 \text{ m}^2$ .

9. [1pt] Consider the reaction of nitrous oxide  $\text{N}_2\text{O}$  with oxygen  $\text{O}_2$  to form nitrogen dioxide  $\text{NO}_2$ :



A 1.0 liter vessel contains  $\text{N}_2\text{O}$  gas at 2.0 atm pressure. A second vessel, 5.0 L in volume contains oxygen at 2.0 atm pressure. Now suppose that the two vessels are connected by a pipe of negligible volume and the two gases mix and react to produce as much nitrogen dioxide as possible. Assume that the temperature remains constant. What is the pressure in the apparatus at the end of the reaction?

10. [2pt] Match each person with the most appropriate description. (If the first corresponds to B, and the next 6 to C, enter BCCCCC)

- |                       |                   |
|-----------------------|-------------------|
| 1) Elizabeth Browning | A. Politician     |
| 2) Leonardo DaVinci   | B. Painter        |
| 3) Socrates           | C. Philanthropist |
| 4) John Keats         | D. Poet           |
| 5) Emanuel Kant       | E. Philosopher    |
| 6) Andrew Mellon      |                   |
| 7) Harry S. Truman    |                   |

11. [2pt] List the following in order of increasing lengths from shortest to longest. (If B is shortest, then A, then C, and D is longest, enter BACD)

- A) 40 mm
- B) 1.00 inch
- C) 0.10 ft
- D) 0.060 m

12. [2pt] Which of the following 20th century vice presidents served later as president of the United States? Give all correct answers (A, CE, BDF, etc.).

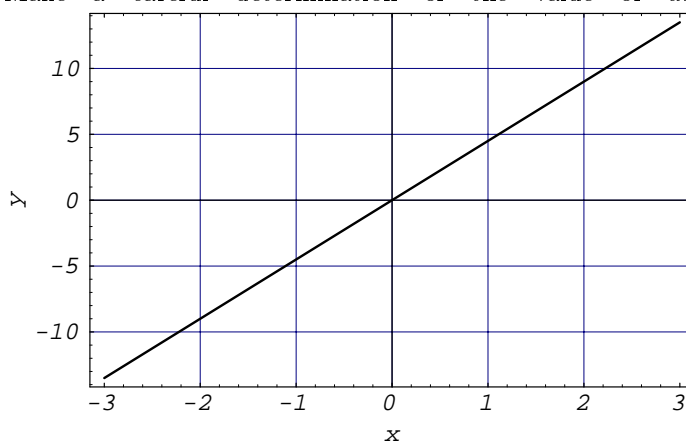
- A) James S. Sherman (NY)
- B) Thomas R. Marshall (IN)
- C) Lyndon B. Johnson (TX)
- D) Walter Mondale (MN)
- E) Harry S. Truman (MO)
- F) Calvin Coolidge (VT)

13. [1pt] Use the 1:1 scale given on your paper and estimate the thickness of the paper used in your textbook.



14. [1pt] The graph shows the function  $Y = ax$

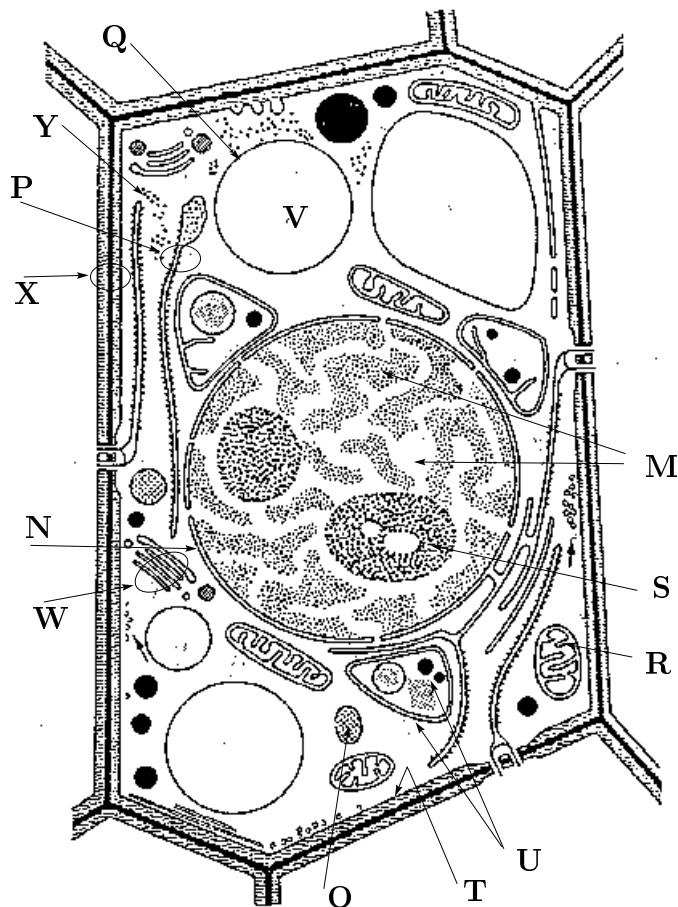
Make a careful determination of the value of  $a$ .



15. [2pt] A fisherman and his young nephew are in a boat on a small pond. Both are wearing life jackets. The nephew is holding a large floating helium filled balloon by a string. Consider each action below independently, and indicate whether the level of the water in the pond R-Rises, F-Falls, S-Stays the Same, C-Can't tell. (If in the first the level Rises, and the in second it Falls, and for the rest one Cannot tell, enter RFCCC)

- A) The nephew pops the balloon.
- B) The nephew gets in the water, loses his grip on the string, letting the balloon escape upwards.
- C) The fisherman fills a glass with water from the pond and drinks it.
- D) The fisherman lowers himself in the water and floats on his back
- E) The fisherman knocks the tackle box overboard and it sinks to the bottom.

16. [3pt] Match the appropriate letter on the diagram with each organelle in the sequence in which they are listed. (Example: If the first organelle corresponds to D on the diagram and the next to C, begin your answer with AC...)

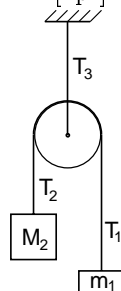


- 1) Peroxisome
- 2) Cell or Plasma Membrane
- 3) Golgi Body
- 4) Plastid (Chloroplast, proplastid, ...)
- 5) Vacuole
- 6) Tonoplast or Vacuolar Membrane
- 7) Nucleus
- 8) Mitochondrion
- 9) Ribosomes
- 10) Cell Wall
- 11) Nuclear Envelope
- 12) Endoplasmic Reticulum
- 13) Nucleolus

17. [1pt] A block is at rest on an inclined plane whose elevation can be varied. The angle of elevation  $\theta$  is increased slowly from the horizontal, and when it reaches 43.1 degrees, the block begins to slide. Calculate the coefficient of static friction.

18. [1pt] The block reaches a speed of  $3.21 \text{ m/s}$  in a time of  $1.12 \text{ s}$ . Calculate the coefficient of kinetic friction. Use  $g=9.81 \text{ m/s}^2$

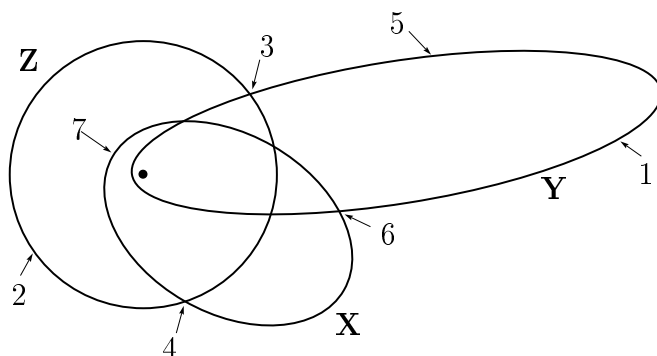
19. [2pt]



A frictionless, massless pulley is attached to the ceiling, in a gravity field of  $9.81 \text{ m/s}^2$ . Mass  $M_2$  is greater than mass  $m_1$ . The quantities  $T_n$  and  $g$  are magnitudes. (For each statement select T -True, F -False, G -Greater than, L -Less than, or E -Equal to. )

- A) The center-of-mass does not accelerate.
- B)  $T_3$  is ....  $m_1g + M_2g$
- C) The magnitude of the acceleration of  $M_2$  is ... the magnitude of the acceleration of  $m_1$ .
- D)  $T_1$  is ....  $T_2$
- E)  $T_1 + T_2$  is ....  $T_3$
- F)  $M_2g$  is ...  $T_2$

20. [2pt] Asteroids X, Y, Z have equal mass ( $4 \text{ kg}$  each). They orbit around a planet with  $M=4 \times 10^{24} \text{ kg}$ . The orbits are in the plane of the paper and are drawn to scale.



TE, KE and PE represent Total, Kinetic and Potential energies. Select G-Greater than, L-Less than, or E-Equal to. (If the first is G and the rest L, enter GLLLLLL).

- A) The speed of Y at 5 is .... it is at 3.
- B) The TE of Z is .... the TE of X.
- C) The PE of Y at 3 is .... the PE of Z at 3.
- D) The PE of Y at 6 is .... the PE of X at 4.
- E) The PE of X at 7 is .... its value at 6.
- F) The KE of Z at 2 is .... its value at 3.
- G) The TE of Y is .... the TE of X.

21. [2pt] The three asteroids orbit in the same direction. Select G-Greater than, L-Less than, or E-Equal to. (If the first is G and the rest E, enter GEEEEEE).

- A) At 3, Z's angular velocity is .... that at 4.
- B) At 4, Z's angular velocity is .... that of X.
- C) Z's angular momentum is .... that of X.
- D) The period of X is .... that of Z.
- E) The angular momentum of X at 6 is .... that at 4.
- F) The angular velocity of X at 6 is .... that at 7.
- G) The period of Z is .... that of Y.

## Source Code for some Sample Problems

Note that a few carriage returns which are NOT in the actual set1.qz file have been added so that the text would not be cut-off below.

Beginning of Set1.qz source code.

```
//CAPA system software is copyrighted by Michigan State University.
//By using these materials, the User agrees to:
//1) Protect the source code files from unauthorized copying.
//2) Limit access of the source material to teaching staff.
//3) The User is free to mix, cut and paste, modify, adapt, delete,
// improve, etc. the problems and graphics for his/her own use.
//
//IMP "../Tools/StdMacros"
//IMP "../Tools/StdUnits"
//IMP "HWTop"
//*****
```

Problem 1 is not shown

Below are problems 2, 3, 4

```
//*****
//*****
//BEG prob_val=1
//IMP "../Tools/Problem#"
//By E. Kashy, kashy@nscl.msu.edu, No commercial use
//LET length=random(12.5,23.6,1.0)
//LET width=random(36.,48.,1)
Calculate the perimeter of a rectangle with a length of /DIS(length:1f)/DIS(cm_u)
and a width of /DIS(width:0f)/DIS(mm_u).
/DIS(web("Enter units","Enter units","<a href= /nsc111f7/Links/UnitFmt.html>Enter units</a>")).
/DIS(html("<a href=/nsc111f7/Links/IN_units.html>[Instructor Note: Units.]</a>"))
//LET peri_cm=2*(length+width/10.0)
//HIN The perimeter of a rectangle is the length of its 4 sides. The perimeter is a length, and units are required.
//EXP Add the length and width and multiply by 2.
//ANS(peri_cm:1f,tol=0.09,wgt=prob_val,tries=10,unit="cm")
//DIS(stdline)
//*****
//*****
//BEG prob_val=1
//IMP "../Tools/Problem#"
//By E. Kashy, kashy@nscl.msu.edu, No commercial use
//LET length=random(12.5,23.6,1.0)
//LET width=random(36.,48.,1)
Calculate, in /DIS(cm_u), the perimeter of a rectangle with a length of /DIS(length:1f)/DIS(cm_u) and
a width of /DIS(width:0f)/DIS(mm_u).
/DIS(html("<a href=/nsc111f7/Links/IN_tries.html>[Instructor Note: Number of tries.]</a>"))
//LET peri_cm=2*(length+width/10.0)
//HIN The perimeter of a rectangle is the sum of the lengths of its 4 sides. Since the answer required is in cm,
//HIN the answer is just a number.
//EXP Add the length and width and multiply by 2.
//ANS(peri_cm:1f,tol=0.09,wgt=prob_val,tries=5)
//DIS(stdline)
//*****
//*****
//BEG prob_val=1
//IMP "../Tools/Problem#"
//*****
```

```
//By E. Kashy, kashy@nscl.msu.edu, No commercial use
/LET length=random(12.5,23.6,1.0)
/LET width=random(36.0,48.0,1.0)
Calculate the perimeter of a rectangle with a length of /DIS(length:2f)/DIS(cm_u) and
a width of /DIS(width:1f)/DIS(mm_u).
/DIS(html("<a href=/nsc111f7/Links/IN_SigFigs.html>[Instructor Note: Significant Figures.]</a>"))
/LET peri_cm=2*(length+width/10.0)
/HIN The perimeter of a rectangle is the length of its 4 sides.
/HIN Careful with units AND with the number of significant figures.
/EXP Add the length and width and multiply by 2.
/ANS(peri_cm:2f,tol=0.009,wgt=prob_val,tries=10,unit="cm",sig=4)
/DIS(stdline)
//*****
```

Below are problems 9, 10, 11, 12  
 Problems 10, 11 and 12 use pre-coded templates.

```
//*****
//*****
/LET prob_val=1
/IMP "../Tools/Problem#"
//a limiting reagent problem for a gas phase reaction
//from p.w.w.hunter
//
/LET jan1=random(1.0,6.5,1.0)
/LET jan2=random(1.0,6.5,1.0)
/LET jan3=random(1.0,6.5,1.0)
/LET jan4=random(1.0,6.5,1.0)
/LET n2o=web("N2O", "N$_2$O", "N<sub>2</sub>O")
/LET o2=web("O2", "O$_2$", "O<sub>2</sub>")
/LET no2=web("NO2", "NO$_2$", "NO<sub>2</sub>")
/LET rxnasc="2N2O(g) + 3O2(g)--> 4NO2(g)"
/LET rxntex="$$ \rm 2N_{20}(g) + 3O_{2}(g) \rightarrow 4NO_{2}(g)$$"
/LET rxnhtml="<center>2N<sub>2</sub><sub>0</sub>(g) + 3O<sub>2</sub>(g) --> 4NO<sub>2</sub>(g)</center>"
Consider the reaction of nitrous oxide /DIS(n2o) with oxygen /DIS(o2) to form nitrogen dioxide /DIS(no2):
/DIS(web(rxnasc,rxntex,rxnhtml)) A /DIS(jan1:1f) liter vessel contains /DIS(n2o) gas at
/DIS(jan2:1f) /DIS(atm_u) pressure.
A second vessel, /DIS(jan3:1f) L in volume contains oxygen at /DIS(jan4:1f) /DIS(atm_u) pressure.
Now suppose that the two vessels are connected by a pipe of negligible volume and
the two gases mix and react to produce as much nitrogen dioxide as possible.
Assume that the temperature remains constant. What is the pressure in the apparatus at the end of the reaction?
/HIN First determine the limiting reagent.
/HIN The limiting reagent will determine how far the reaction can possibly go.
/HIN Don't forget to enter the appropriate unit for pressure.
/LET pn=(jan2)*(jan1)/(jan1+jan3)
/LET po=(jan4)*(jan3)/(jan1+jan3)
/LET terra=((pn/2)>(po/3))+1)
/LET ansn=(pn*2)+(po-3*pn/2)
/LET anso=(po*4/3)+(pn-2*po/3)
/LET ans10=choose(terra,ansn,anso)
/ANS(ans10:2e,sig=3 plus 2 minus 2,tol=3%,wgt=prob_val,unit="atm",tries=10)
/DIS(stdline)
//*****
//*****
/BEG prob_val=2
/IMP "../Tools/Problem#"
//By E. Kashy, kashy@nscl.msu.edu, No commercial use
// 7 items, 5 properties. Problems Text follows the next line directly.
Match each person with the most appropriate description. (If the first corresponds to B, and
```

```

the next 6 to C, enter BCCCCC)
/LET tp1="Poet"
/LET tp2="Painter"
/LET tp3="Politician"
/LET tp4="Philosopher"
/LET tp5="Philanthropist"
//
/LET s1a="Edgar Allen Poe" // - The raven - For Annie - Annabel Lee
/LET s1b="Elizabeth Browning" //Sonnets from the portuguese Aurora Leigh
/LET s1c="Statement 1 variation c"
/LET s1d="Statement 1 variation d"
/LET mix1=random(1,2,1)
/LET x1a=1
/LET x1b=1
/LET x1c=26
/LET x1d=26
//
/LET s2a="John Keats" // -Ode on Grecian Urn - Ode to a nightingale
/LET s2b="Percy Shelly" // -To a skylark - Prometheus unbound
/LET s2c="Statement 2 variation c"
/LET s2d="Statement 2 variation d"
/LET mix2=random(1,2,1)
/LET x2a=1
/LET x2b=1
/LET x2c=26
/LET x2d=26
//
/LET s3a="Leonardo DaVinci"
/LET s3b="Claude Monet"
/LET s3c="Statement 3 variation c"
/LET s3d="Statement 3 variation d"
/LET mix3=random(1,2,1)
/LET x3a=2
/LET x3b=2
/LET x3c=26
/LET x3d=26
//
/LET s4a="Emanuel Kant" //1724-1804
/LET s4b="Plato" //427-347BC
/LET s4c="Statement 4 variation c"
/LET s4d="Statement 4 variation d"
/LET mix4=random(1,2,1)
/LET x4a=4
/LET x4b=4
/LET x4c=26
/LET x4d=26
//
/LET s5a="Andrew Mellon" //- National Gallery of Art, in DC -
/LET s5b="Alfred P. Sloan" //GM president- Sloan-Kettering cancer Inst.
/LET s5c="Statement 5 variation c"
/LET s5d="Statement 5 variation d"
/LET mix5=random(1,2,1)
/LET x5a=5
/LET x5b=5
/LET x5c=26
/LET x5d=26
//
/LET s6a="Socrates" //470-399 "Know thyself" Known thru Plato's writings
/LET s6b="Statement 6 variation b"

```



```

/LET s6c="Statement 6 variation c"
/LET s6d="Statement 6 variation d"
/LET mix6=random(1,1,1)
/LET x6a=4
/LET x6b=26
/LET x6c=26
/LET x6d=26
//
/LET s7a="Harry S. Truman"
/LET s7b="Woodrow Wilson"
/LET s7c="Sam Rayburn"
/LET s7d="Richard M. Nixon"
/LET mix7=random(1,4,1)
/LET x7a=3
/LET x7b=3
/LET x7c=3
/LET x7d=3
/IMP "../MCTools/i7p5auxw"
/ANS(i7p5right,wgt=prob_val,tries=10)
/DIS(stdline)
/*****
/*****
/LET prob_val=2
//HIN No Hint unless you un-comment this text to make it into a hint.
//EXP No Explanation unless you un-comment this text and write something.
/IMP "../Tools/Problem#"
//4 Statements. Rank4 tool (see Rank 5 for more variations of each item)
List the following in order of increasing lengths from shortest to longest. (If B is shortest,
then A, then C, and D is longest, enter BACD )
// FILL Qty1, Qty2,.. as strings; assign a relative size rs1,rs2,..
//
/LET Qty1="0.10 ft"
/LET Qty2="40 mm"
/LET Qty3="0.060 m"
/LET Qty4="1.00 inch"
// The values below are all in mm
/LET rs1=0.1*12*25.4
/LET rs2=40.0
/LET rs3=60.0
/LET rs4=25.4
/IMP "../MCTools/Rank4aux"
/ANS(Rank4right,wgt=prob_val,tries=10)
/DIS(stdline)
/*****
/*****
/BEG prob_val=2
/IMP "../Tools/Problem#"
//By E. Kashy, kashy@nscl.msu.edu, No commercial use
// ../MCTools/Nof8 Select N correct of 8 Statements
Which of the following 20th century vice presidents served later as president of the United States?
Give all correct answers (A, CE, BDF, etc.).
/DIS(html("<a href=/nsc111f7/Links/IN_CodeTools.html>[Instructor Note: Coding Tools.]</a>"))
// set ala = 1 for a correct variation, and 2 for incorrect, etc..
// ALL variations for at least one statement must always be correct!
//-----
/LET s1a="Theodore Roosevelt (NY)"
/LET s1b="Lyndon B. Johnson (TX)"
/LET s1c="Statement 1 variation c"
/LET s1d="Statement 1 variation d"

```

```

/LET mix1=random(1,2,1)
/LET a1a=1
/LET a1b=1
/LET a1c=26
/LET a1d=26
//
/LET s2a="Calvin Coolidge (VT)"
/LET s2b="Statement 2 variation b"
/LET s2c="Statement 2 variation c"
/LET s2d="Statement 2 variation d"
/LET mix2=random(1,1,1)
/LET a2a=1
/LET a2b=26
/LET a2c=26
/LET a2d=26
//
/LET s3a="Harry S. Truman (MO)"
/LET s3b="Statement 3 variation b"
/LET s3c="Statement 3 variation c"
/LET s3d="Statement 3 variation d"
/LET mix3=random(1,1,1)
/LET a3a=1
/LET a3b=26
/LET a3c=26
/LET a3d=26
//
/LET s4a="Thomas R. Marshall (IN)"
/LET s4b="George Bush (MA)"
/LET s4c="Statement 4 variation c"
/LET s4d="Statement 4 variation d"
/LET mix4=random(1,2,1)
/LET a4a=2
/LET a4b=1
/LET a4c=26
/LET a4d=26
//
/LET s5a="James S. Sherman (NY)"
/LET s5b="Statement 5 variation b"
/LET s5c="Statement 5 variation c"
/LET s5d="Statement 5 variation d"
/LET mix5=random(1,1,1)
/LET a5a=2
/LET a5b=26
/LET a5c=26
/LET a5d=26
//
/LET s6a="Walter Mondale (MN)"
/LET s6b="Statement 6 variation b"
/LET s6c="Statement 6 variation c"
/LET s6d="Statement 6 variation d"
/LET mix6=random(1,1,1)
/LET a6a=2
/LET a6b=26
/LET a6c=26
/LET a6d=26
//
/IMP "../MCTools/Nof6aux"
/ANS(Nof6right,wgt=prob_val,str=mc,tries=10)
/DIS(stdline)

```

```
//*****
```

Below are problems 14, 17, 18

```
//*****
```

```
/BEG prob_val=1
/IMP "../Tools/Problem#"
//By E. Kashy, kashy@nscl.msu.edu, No commercial use
/LET k=random(1,14,1)
/LET file=choose(k,"35","45","56","62","77","86","93","m26","m39","m47","m51","m66","m74","m84")
/LET ans5=choose(k, 3.5,4.5,5.6,6.2,7.7,8.6,9.3,-2.6,-3.9,-4.7,-5.1,-6.6,-7.4,-8.4)
The graph shows the function /DIS(tex("\ \centerline{\$Y=a x\$} ", "Y=ax. ")) Make a careful determination of
the value of /DIS(tex("\$a\$", "a")).
/DIS(tex("\epsfxsize=3.6in \epsffile{/usr/users/teacher/CAPA45/nsc111f7/pictures/Y", ""))
/DIS(var_in_tex(file))/DIS(tex("x.ps", ""))
/DIS( html("<IMG SRC=/nsc111f7/pictures/Y") )/DIS(html(file))/DIS(html("x.gif >") )
/ANS(ans5:2E,tol=5%,wgt=prob_val,tries=10)
/DIS(stdline)
```

```
//*****
```

```
//*****
```

```
/BEG prob_val=1
/IMP "../Tools/Problem#"
//By E. Kashy kashy@nscl.msu.edu No unauthorized commercial use.
/LET thetaslide=random(34.1,43.6,0.2)
A block is at rest on an inclined plane whose elevation can be varied. The angle of elevation
/DIS(web("theta", "\theta", "<b><i>theta</i></b>")) is increased slowly from the horizontal,
and when it reaches /DIS(thetaslide:1f) degrees, the block begins to slide.
Calculate the coefficient of static friction.
/DIS(html("<a href=/nsc111f7/Links/TFCanim1.html>Motion of a block</a>"))
/LET pi=3.141592654
/LET thetaRad=thetaslide*pi/180.0
/LET muStat=tan(thetaRad)
/ANS(muStat:2E,tol=1%,wgt=prob_val, tries=10)
//-----
//-----
```

```
/BEG prob_val=1
/IMP "../Tools/Problem#"
//By E. Kashy kashy@nscl.msu.edu No unauthorized commercial use.
/LET slidetime=random(1.12, 1.475, 0.01)
/LET speed=random(2.87, 3.435, 0.01)
The block reaches a speed of /DIS(speed:2f)/DIS(mPs_u) in a time of /DIS(slidetime:2f)/DIS(s_u).
Calculate the coefficient of kinetic friction. Use  $g=9.81/\text{DIS}(mPs2_u)$ 
/HIN The block accelerates because  $\mu_k$  is less than  $\mu_s$ 
/LET accel=speed/slidetime
/LET muk=((9.81*sin(thetaRad)-accel))/(9.81*cos(thetaRad))
/ANS(muk:2E,tol=1%,wgt=prob_val, tries=10)
/DIS(stdline)/DIS(newpage)
//*****
```

Below is problems 19

```
//*****
```

```
/BEG prob_val=2
/IMP "../Tools/parProblem#"
/HIN This problem deals with Newton's second law,  $F=ma$ . Think about acceleration and the net force on 'a body'.
A 'body' can be a set of objects connected by internal forces.
// by E.Kashy, kashy@nscl.msu.edu No unauthorized commercial
//use.
```

```

// ../CAPA/MCTools/M6T7 6 items matched to up to 7 strings defined
//in the problem statement. The strings are sT1, sT2, sT3, sT4, sT5, sT6, sT7.
// Fill out symbols (sTx) and then meanings (sTrx)
/LET sT1="T" // reference value is 1, could have used "Tr"
/LET sT2="F" // reference value is 2, could have used "Fa"
/LET sT3="G" // reference value is 3, could have used tex("$>$",">")
/LET sT4="L" // reference value is 4, could have used tex("$<$","<")
/LET sT5="E" // reference value is 5. etc.
/LET sT6="H" // reference value is 6
/LET sT7="B" // reference value is 7
//Fill out meanings;
/LET sTr1="-True, " // If only 2 symbols are needed , use this one and
/LET sTr2="-False, " // this one. Note the dash and comma and space here.
/LET sTr3="-Greater than, " // Leaving blank for the sTx and sTrx strings
/LET sTr4="-Less than, " // will automatically format the instructions
/LET sTr5="-Equal to. " // for the students.
/LET sTr6="- Heavy, "
/LET sTr7="-Blue. " // last filled out one has a period.
//
//
/LET mOne=web("m1","$m_1$","m<sub>1</sub>")
/LET MTwo=web("M2","$M_2$","M<sub>2</sub>")
/LET TOne=web("T1","$T_1 \ $","T<sub>1</sub>")
/LET TTwo=web("T2","$T_2 \ $","T<sub>2</sub>")
/LET TThree=web("T3","$T_3 \ $","T<sub>3</sub>")
//
/LET picture1=web("", "\parbox{1 in}{\centerline{\epsfxsize=0.6 in
\hskip -.05in \epsffile{/usr/users/teacher/CAPA45/nsc111f7/pictures/Atwood1.eps}}}",
"<IMG SRC=/nsc111f7/pictures/Atwood1crl.gif ALIGN=left width=95>")
//
/DIS(picture1)/DIS(tex("\parbox{2.4 in}{","")A /DIS(web("frictionless, massless",
"frictionless, massless",
"<a href=/nsc111f7/Links/frictionless.html>frictionless, massless</a>"))
pulley is attached to the ceiling, in a gravity field of 9.81/DIS(mPs2_u).
Mass /DIS(MTwo) is greater than mass /DIS(mOne).
The quantities /DIS(web("Tn","$T_n$","T<sub>n</sub>")) and /DIS(tex("$g$","g")) are magnitudes.
(For each statement select /DIS(sT1) /DIS(sTr1) /DIS(sT2) /DIS(sTr2) /DIS(sT3) /DIS(sTr3) /DIS(sT4) /DIS(sTr4) or
/DIS(sT5) /DIS(sTr5))/DIS(tex("}",""))
/DIS(html("<br><a href=/nsc111f7/Links/webPulley.html>Motion of Masses on a Pulley</a>"))
/DIS(html("<p><br><table><tr><td>"))
//
/LET s1a=TTwo +" is .... "+ TOne
/LET s1b=TOne +" is .... "+ TTwo
/LET s1c="Statement 1 variation c"
/LET s1d="Statement 1 variation d"
/LET mix1=random(1,2,1)
/LET a1a=5
/LET a1b=5
/LET a1c=26
/LET a1d=26
//
/LET s2a="The magnitude of the acceleration of " + MTwo +
" is ... the magnitude of the acceleration of " + MOne + "."
/LET s2b="Statement 2 variation b"
/LET s2c="Statement 2 variation c"
/LET s2d="Statement 2 variation d"
/LET mix2=random(1,1,1)
/LET a2a=5
/LET a2b=26

```

```

/LET a2c=26
/LET a2d=26
//
/LET s3a=TThree + " is ... " + MOne+"g + " + MTwo+"g"
/LET s3b=MOne+"g + " + MTwo+"g is ... "+TThree
/LET s3c="Statement 3 variation c"
/LET s3d="Statement 3 variation d"
/LET mix3=random(1,2,1)
/LET a3a=4
/LET a3b=3
/LET a3c=26
/LET a3d=26
//
/LET s4a=TThree +" is ... "+ TOne+ "+"+TTwo
/LET s4b=TOne+ "+"+TTwo +" is ... "+TThree
/LET s4c="Statement 4 variation c"
/LET s4d="Statement 4 variation d"
/LET mix4=random(1,2,1)
/LET a4a=5
/LET a4b=5
/LET a4c=26
/LET a4d=26
//
/LET s5a="The center-of-mass does not accelerate."
/LET s5b="The center-of-mass accelerates."
/LET s5c="Statement 5 variation c"
/LET s5d="Statement 5 variation d"
/LET mix5=random(1,2,1)
/LET a5a=2
/LET a5b=1
/LET a5c=26
/LET a5d=26
//
/LET s6a=TOne+ " is ... "+MOne+"g"
/LET s6b=TTwo+ " is ... "+MTwo+"g"
/LET s6c=MOne+"g"+ " is ... "+TOne
/LET s6d=MTwo+"g"+ " is ... "+TTwo
/LET mix6=random(1,4,1)
/LET a6a=3
/LET a6b=4
/LET a6c=4
/LET a6d=3
/IMP "../MCTools/M6T7aux"
/DIS(html("</td></tr></table>"))
/ANS(M6T7right,wgt=prob_val,tries=10)
/DIS(stdline)
//*****

```

Problems 20 and 21 are not shown.  
The source code for the complete set is in the distribution package.

```

//*****
/END(stdendline)

```

---

---

### *References and Additional Information*

The *CAPA* homepage URL is <http://www.pa.msu.edu/educ/CAPA/> It includes the document ‘An Introduction to *CAPA*: A Versatile Tool for Science Education’ in postscript format. This document outlines hardware specifications for *CAPA* as well as licensing information.

---

### *Technical Support*

For technical support, email: [nedavis@nextdjm.nscl.msu.edu](mailto:nedavis@nextdjm.nscl.msu.edu)  
or call: (517)333-6396

Other e-mail resources are:  
[kashy@nscl.msu.edu](mailto:kashy@nscl.msu.edu)  
[morrissey@nscl.msu.edu](mailto:morrissey@nscl.msu.edu)

---

### *Acknowledgments*

Many instructors, graduate students, and undergraduate students at MSU and other institutions have participated in the *CAPA* project. Their work and suggestions have helped to make *CAPA* a better tool.

The support for *CAPA* development or operation from the following is gratefully acknowledged. :

- The College of Natural Science
- The Office for Computing and Technology
- The Department of Physics and Astronomy
- The Department of Chemistry
- The Alfred P. Sloan foundation
- The National Superconducting Cyclotron Laboratory(NSCL)
- The US Department of Agriculture
- The MSU Provost

The advice of R. Fox, K. Berhooz and B. Pollack of the (NSCL) and G. Perkins (of the Department of Physics and Astronomy) has been of considerable help in computer and network operation.

---

*CAPA* license can be obtained from:

Instructional Media Center  
Michigan State University  
East Lansing, MI 48826-0710

See also: <http://www.pa.msu.edu/educ/CAPA/>