24780 Engineering Computation: Problem Set 2

(*) In the following instructions (and in all course materials), substitute your Andrew ID wherever you see *yourAndrewId*.

You need to create a ZIP file (which may appear as a compressed folder in Windows) and submit the ZIP file via the 24-780 Canvas. The filename of the ZIP file must be:

```
PS02-YourAndrewID.zip
```

For example, if your Andrew account is hummingbird@andrew.cmu.edu, the filename must be:

```
PS02-hummingbird.zip
```

Failure to comply with this naming rule will result in an automatic 5% deduction from this assignment's credit. If we cannot identify the submitter of the file, an additional 5% credit will be lost. If we are ultimately unable to connect you with the submitted ZIP file, you will receive 0 points for this assignment. Therefore, ensure strict adherence to this naming rule before submitting a file.

The ZIP file must be submitted to the 24-780 Canvas. If you find a mistake in a previous submission, you can re-submit the ZIP file with no penalty as long as it's before the submission deadline.

Notice: The grade will be assigned to the final submission only. In the case of multiple file submissions, earlier versions will be discarded. Therefore, when resubmitting a ZIP file, it MUST include all the required files. Also, if your final version is submitted after the submission deadline, the late-submission policy will be applied, regardless of how early your earlier version was submitted.

Ensure that your program can be compiled without errors on one of the compiler servers. Do not wait until the last minute, as the compiler servers may become very busy just minutes before the submission deadline!

Submission Due: Please refer to Canvas.

PS2-1 Weekly Lunch Planner [ps2-1.cpp] (30 pts)

Decisions, decisions! Everyday, we need to make difficult decisions including what to eat for lunch. Famous physicist, Dr. Richard Feynman solved this problem by eating same thing for lunch every day. But, we need to eat balanced food to stay healthy. So, how about letting the computer decide what to eat for lunch? Make a list of seven different candidates for your lunch (hamburger, pizza, sub, ramen noodle, etc.). Use shuffling to randomize the order of food, and present on the console window like:

```
Sun Pizza from Mineo's
Mon Tofu Tikka Masala from Prince of India
Tue Crispy Salmon Roll from Sushi Fuku
Wed Sub from Uncle Sam's
Thr Fried rice from How Lee
Fri Sandwiches from La Prima
Sat Find free food on campus
```

Save your source file as ps2-1.cpp and include in the zip file you submit to Canvas.

PS2-2 Digital Flashcards of Multiplication [ps2-2.cpp] (70 pts)

In this problem, you will write a C++ code that serves as a learning tool for elementary multiplication covering the range from 1x1 to 12x12. Your C++ code should be a console application and roughly mimic the functionality of a conventional paper flashcard, shown in Fig. 1.

A flashcard is any of a set of cards bearing information, as words of numbers, on either or both sides, used in classroom drills or in private study. Flashcards can bear vocabulary, historical dates, formulas or any subject matter that can be learned via a question and answer format. Flashcards are widely used as a learning drill to aid memorization by way of spaced repetition.

Your C++ console flash card should:

- First ask the user how many flash cards to work on. The number needs to be between 1 and 144. If the user requests less than 1 or more than 144 flash cards, show a message (eg. "The number of cards must be between 1 and 144.") and prompt the user to re-enter a different number.
- Show questions one by one on the console window and prompt the user to type the answer. If the user types a wrong answer, tell the user that the answer is incorrect and display the correct answer and then move on to the next card. If the answer is correct, tell the user that the answer is correct, and then move on to the next card.
- When all the flash cards have been answered, display a feedback comment that includes:
  - The number of problems,
  - Time required to complete the problems in the number of seconds, and
  - The number of correct answers and percent that the user answered correctly.

  For example, "You answered 25 problems in 80 seconds. You answered 20 problems correctly (80%)"
- Problems should be selected and presented randomly, but no question should appear more than once. (X times Y and Y times X are considered two different questions since elementary-school kids need to learn X*Y and Y*X give the same answer.)

The card should be presented as follows:

```
+-----+
|     |
|12x3 |
|     |
+-----+
Enter Your Answer>
```

It should be 7 characters wide and 5 lines high. The question needs to be in the middle line. When one or both numbers are less than 10, add space to align the right vertical bar. Spaces can be added anywhere as long as they do not split a number. Therefore,

```
+-----+ +-----+ +-----+ +-----+ +-----+ +-----+ +-----+ +-----+ +-----+
|     | |     | |     | |     | |     | |     | |     | |     | |     |
|12x 3| |11x 5| | 10x4| |3x8  | |  6x4| | 8x 6| |2 x3 | | 5x3 | |7 x 7|
|     | |     | |     | |     | |     | |     | |     | |     | |     |
+-----+ +-----+ +-----+ +-----+ +-----+ +-----+ +-----+ +-----+ +-----+
```

are all acceptable, but:

```
+-----+
|     |
|1 2x3|
|     |
+-----+
```

is not acceptable because it splits the number 12.

Text does not have to be exact as long as it makes sense to the user. Points will not be deducted for minor English mistakes.

Fig. 2 shows a sample screenshot of the program.

Write the C++ program named ps2-2.cpp.

The flashcards should be an array of integers. Give each card a number so that higher 2 digits and lower 2 digits are the two numbers on the card. For example, card number 1107 is for 11 times 7.

Make shuffling a separate function and use it for randomizing cards. Make your shuffling function re-usable in the future, i.e., the function needs to take two parameters, one is number of elements in the array, and the other is an array of integers. Your shuffling function should work for any number of integers, not just for 144 integers.

The .cpp file must be included in the Zip file submitted to Canvas. But, do not include auto-generated project files.

Your Zip file should contain only two files, ps2-1.cpp and ps2-2.cpp. Do not include project files and intermediate files generated by the compiler.

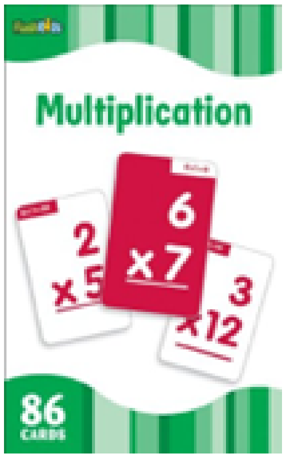
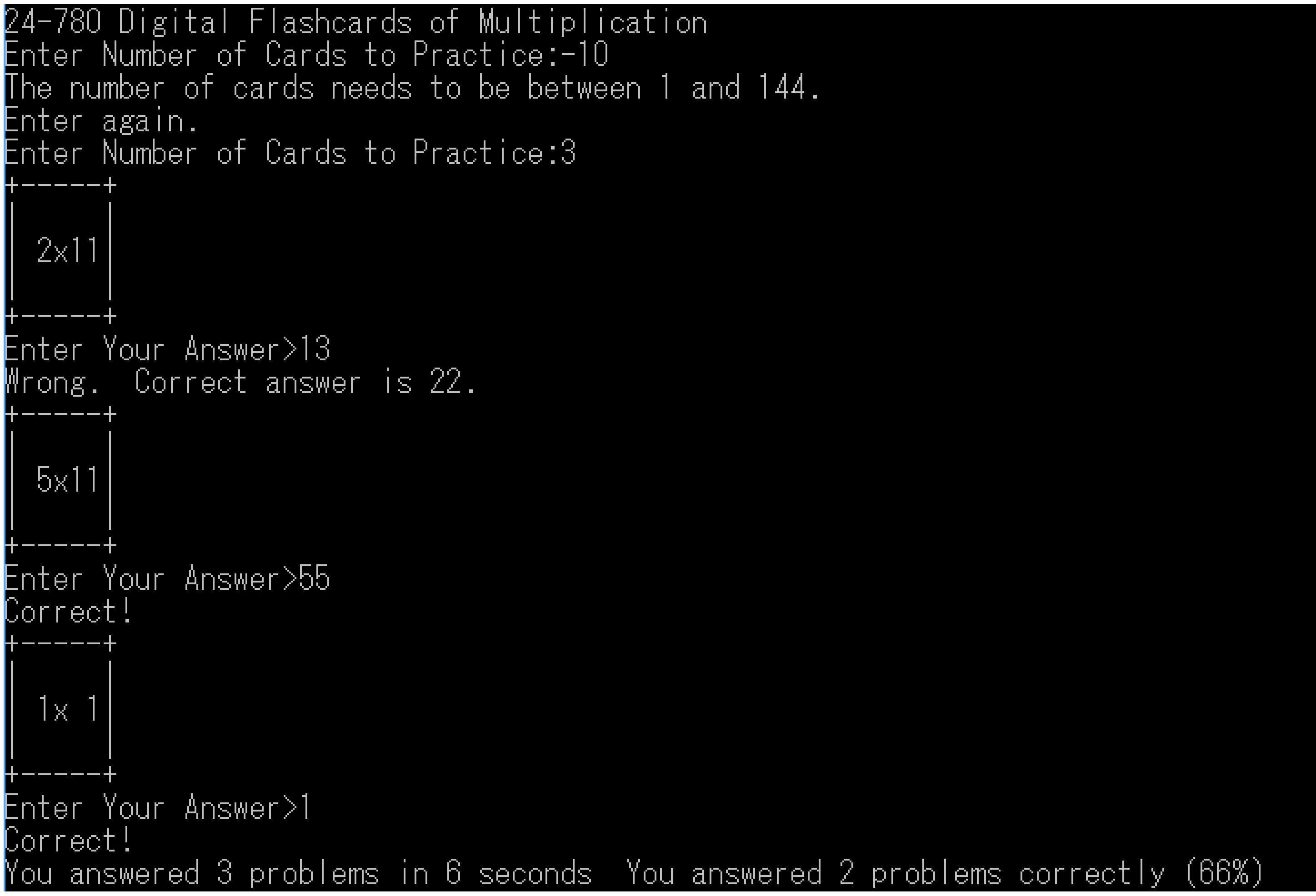


Fig. 1: Flashcard Example



```
24-780 Digital Flashcards of Multiplication
Enter Number of Cards to Practice:-10
The number of cards needs to be between 1 and 144.
Enter again.
Enter Number of Cards to Practice:3
+------+
|      |
| 2x11 |
|      |
+------+
Enter Your Answer>13
Wrong.  Correct answer is 22.
+------+
|      |
| 5x11 |
|      |
+------+
Enter Your Answer>55
Correct!
+------+
|      |
| 1x 1 |
|      |
+------+
Enter Your Answer>1
Correct!
You answered 3 problems in 6 seconds  You answered 2 problems correctly (66%)
```

Fig. 2: Sample Screenshot

Test Your Program with One of the Compiler Servers

Test your program with one of the following compiler servers:

```
http://freefood1.lan.local.cmu.edu
http://freefood2.lan.local.cmu.edu
http://freefood3.lan.local.cmu.edu
http://freefood4.lan.local.cmu.edu
```

You need to make sure you are not getting any errors (red lines) from the compiler server.

It is a good practice to remove warnings as well. However, we will not take points off for warnings as long as your program satisfies requirements of the assignment.

You can only access these servers from CMU network. If you need to access from your home, use CMU VPN. Please visit the CMU computing services web site how to install the VPN.