Small Program 4

COP3223C Introduction to Programming with C Fall 2022
Dr. Andrew Steinberg

Due Date

The assignment is due on October 17th at 11:59pm EST via Webcourses. **Do not email the professor or TAs your submissions as they will not be accepted!** This assignment is accepted late up to 24 hours with a penalty. Please see the syllabus for more information on this. Make sure to submit on time to get potential full credit. Make sure to also take into consideration the uploading time. In the past, students who are working last minute on the assignment sometimes run into uploading issues where their Internet may run slow, resulting in late submissions. The timestamp Webcourses uses for your submission will be applied and will be the final say. Please do not email the instructor or TAs saying your Internet was running slow. If the time is off by a second of the due date, then the assignment is considered late. Plan accordingly!

Important! Read Carefully!

This assignment contains a set of problems that are to be completed in **one C file**. You have learned about creating user-defined functions and why they are so beneficial to us programmers. For each problem in the assignment, you will create the definition of the user-defined function that is asked in the description. **If you do not create a user-defined function for each of the problems, then you will receive no credit for the problem.** Creating user-defined functions is good practice! You also must write the function prototypes! Missing function prototypes will result in points being deducted. Function prototypes are also good practice as well. The file must be named *smallprogram4_lastname_firstname.c*, where lastname and firstname is your last and first name (as registered in webcourses). For example Dr. Steinberg's file would be named *smallprogram4_Steinberg_Andrew.c*. Make sure to include the underscore character _. If your file is not named properly, points will be deducted. The script the graders use will pull your name from the file name properly. It is imperative you follow these steps so you can get your points!

Testing on Eustis

It is your responsibility to test your code on Eustis. If you submit your assignment without testing on Eustis, you risk points being deducted for things that may behave differently on your operating system. Remember, you cannot dispute grades if your code didn't work properly on

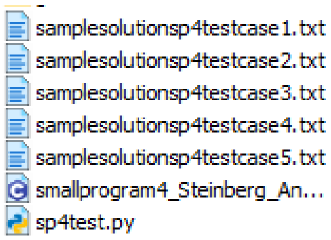
Eustis all because it worked on your machine. The Eustis environment gives the final say. Plan accordingly to test on Eustis!!

Displaying Statements! Read Carefully!

Dr. Steinberg and his TAs have noticed that students are not using the escape sequence newline for the last display statement of past small programs. Please make sure to use the newline escape character for all statements. The only time you will not use the newline escape character is when user input is collected. Pay close attention to the screenshots.

The Python Script File! Read Carefully! NEW INFORMATION!

A python script has been provided for you to test your code with a sample output of Dr. Steinberg's solution. This script will check to make sure your output matches exactly with Dr. Steinberg's solution file as the graders are using this to grade your assignments. The script removes leading and trailing white space, but not white space in the actual text. If there is anything off with the output (including the expected answer), the script will say your output is not correct. This includes your output producing the correct answer, however there is something off with the output display. New Info: The script is going to run 5 unique scenarios for each problem (5 Test Cases). Each test case contains a different set of input values being used to ensure your code produces the correct answer. Back in your previous assignments, Dr. Steinberg would provide 1 sample solution that you would upload to Eustis. Now, there are 5 solution text files you are going to need to upload to Eustis. Before you test your program, your directory in Eustis should look something like this: After you run the script, 5 new text files are going to be generated.

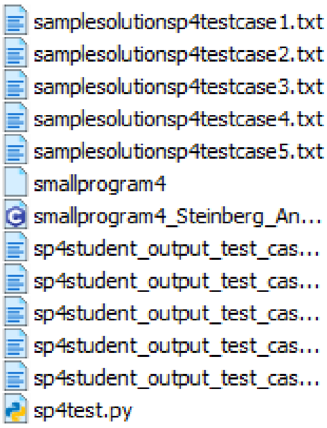


samplesolutionsp4testcase1.txt
samplesolutionsp4testcase2.txt
samplesolutionsp4testcase3.txt
samplesolutionsp4testcase4.txt
samplesolutionsp4testcase5.txt
smallprogram4_Steinberg_An...
sp4test.py

Figure 1: Your setup for testing on Eustis. 5 sample txt files (provided for you in Webcourses), your C program, and the python test script.

These files are the solution output for each test case. If you have these files, you are ready to run the script. Use the following command to test your code with Dr. Steinberg's provided solution sample.

```
python3 sp4test.py
```



samplesolutionsp4testcase1.txt
samplesolutionsp4testcase2.txt
samplesolutionsp4testcase3.txt
samplesolutionsp4testcase4.txt
samplesolutionsp4testcase5.txt
smallprogram4
smallprogram4_Steinberg_An...
sp4student_output_test_cas...
sp4student_output_test_cas...
sp4student_output_test_cas...
sp4student_output_test_cas...
sp4student_output_test_cas...
sp4test.py

Figure 2: Your Eustis setup after running the script in Eustis.

The Rubric

Please see the assignment page for the established rubric on webcourses.

Comment Header

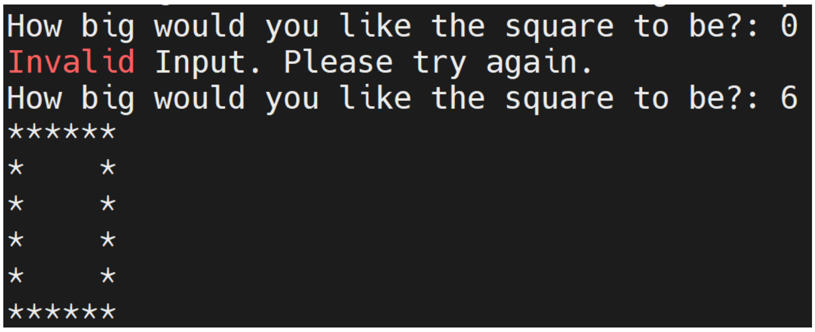
Make sure you place a comment header at the top of your C file. You will use single line comments to write your name, professor, course, and assignment. For example, Dr. Steinberg's header would be:

```
//Andrew Steinberg
//Dr. Steinberg
//COP3223C Section 1
//Small Program 4
```

Missing a comment header will result in point deductions!

Problem 1

Write a user-defined function definition called `perfectSquare` that prints a nice hollow square made out of * characters. The function has no parameters and does not return any values. Inside the function definition, you will prompt the user for a number that will be used in generating the square. If the user inputs an invalid number (0 or negative), then an error message should display informing the user and to try again. See figure 3 for a sample output of the problem.

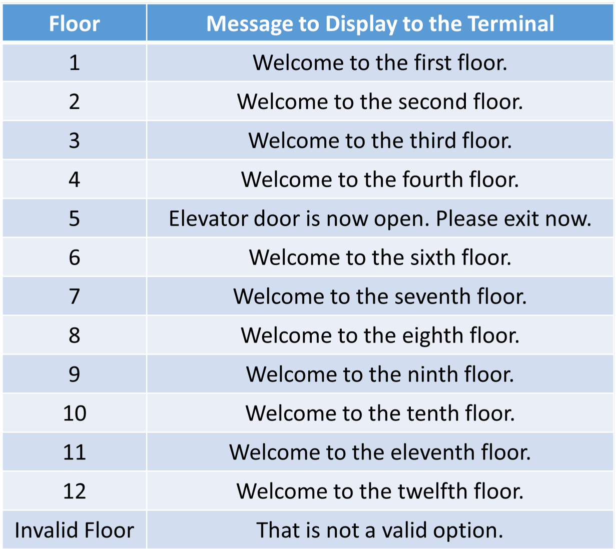


```
How big would you like the square to be?: 0
Invalid Input. Please try again.
How big would you like the square to be?: 6
******
*    *
*    *
*    *
*    *
******
```

Figure 3: Sample out for problem 1. Make sure your output matches this for the script!

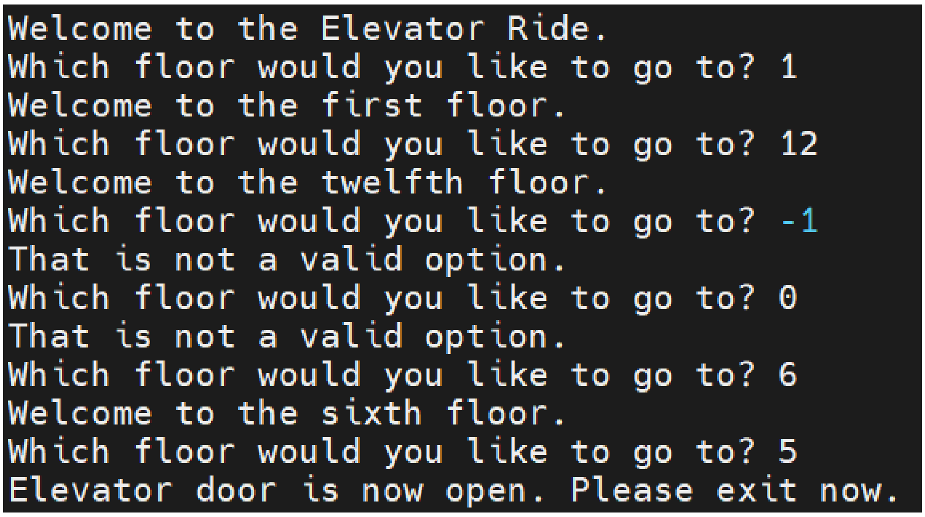
Problem 2

Write a user defined function definition called `elevator` that simulates a menu of an elevator with options the user can select. The function takes no arguments and does not return anything. The user defined function will ask the user which floor they would like to go to. The user will select one of the twelve options and the program will display the floor selected. See figure 4 for the message that is displayed for each respective floor number. If the user selects any valid option 1-12, the program will ask the user to enter another option. If the user selects option 5, the program will not loop again and display the message “Elevator door is now open. Please exit now." If the user selects an invalid option, the message “That is not a valid option." is displayed. Figure 5 shows a sample run on the terminal.



Floor	Message to Display to the Terminal
1	Welcome to the first floor.
2	Welcome to the second floor.
3	Welcome to the third floor.
4	Welcome to the fourth floor.
5	Elevator door is now open. Please exit now.
6	Welcome to the sixth floor.
7	Welcome to the seventh floor.
8	Welcome to the eighth floor.
9	Welcome to the ninth floor.
10	Welcome to the tenth floor.
11	Welcome to the eleventh floor.
12	Welcome to the twelfth floor.
Invalid Floor	That is not a valid option.

Figure 4: Output based on the floor selected.

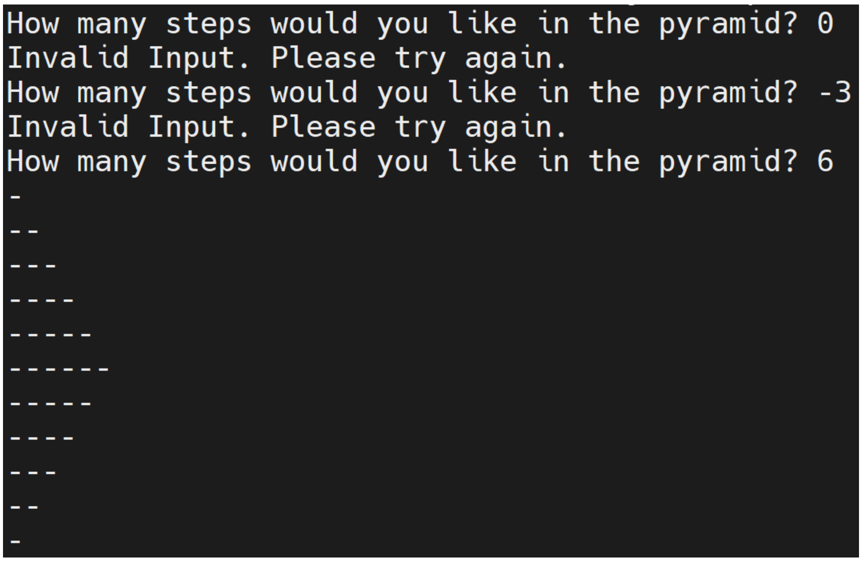


```
Welcome to the Elevator Ride.
Which floor would you like to go to? 1
Welcome to the first floor.
Which floor would you like to go to? 12
Welcome to the twelfth floor.
Which floor would you like to go to? -1
That is not a valid option.
Which floor would you like to go to? 0
That is not a valid option.
Which floor would you like to go to? 6
Welcome to the sixth floor.
Which floor would you like to go to? 5
Elevator door is now open. Please exit now.
```

Figure 5: Sample output for problem 2. Make sure the output matches for the script to test.

Problem 3

Write a user-defined function definition called `pyramid` that prints the following following pattern. The function has no parameters and does not return any values. Inside the function definition, you will prompt the user for a number that will be used in generating the pattern of ‘-’. If the user inputs an invalid number (0 or negative), then an error message should display informing the user and to try again. See figure 6 for a sample output of the problem.

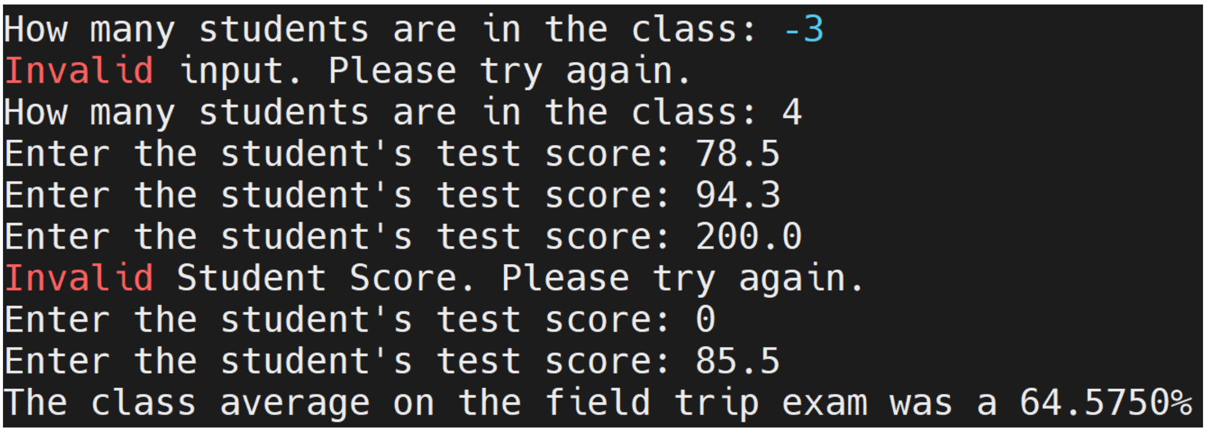


```
How many steps would you like in the pyramid? 0
Invalid Input. Please try again.
How many steps would you like in the pyramid? -3
Invalid Input. Please try again.
How many steps would you like in the pyramid? 6
-
--
---
----
-----
------
-----
----
---
--
-
```

Figure 6: Sample out for problem 3. Make sure your output matches this for the script!

Problem 4

You have been asked by the legendary Ms. Valerie Frizzle to calculate the average of a recent test from a magic school bus field trip in space. Write a user defined function called `classAvg`. The function has one parameter that represents the number of the students in the class. That value is collected in the main function (you also have to assume that an invalid number 0 or a negative number could be entered). If an invalid number is entered, then the user should be asked to enter another value. Once the proper value for the number of students is entered, the function is then going to prompt the user to enter the test score of each student. The scores can range from $0 \leq score \leq 100$ (there can even be decimal values). The function must also handle if the user enters an invalid score that is not in the provided range. If the user enters an invalid score, the program will the user again to input the proper value. Once all scores are entered, the average is computed and sent back to the main function to be displayed. The value of average is of type double. The resulting average is displayed as a percentage up to four decimal places. The following figure shows a sample run with class size of 4. For this problem, you cannot use arrays (we have not covered them yet)! If arrays are used, then No Credit will be given for this problem.



```
How many students are in the class: -3
Invalid input. Please try again.
How many students are in the class: 4
Enter the student's test score: 78.5
Enter the student's test score: 94.3
Enter the student's test score: 200.0
Invalid Student Score. Please try again.
Enter the student's test score: 0
Enter the student's test score: 85.5
The class average on the field trip exam was a 64.5750%
```