Draw Figures **Due: Thursday, Feb 7, 2019, 11:30 pm**

1 Objectives

To practice C basic constructs and C functions.

2 Overview

The first thing you need to do is to copy the directory draw_figures we have left in the grace cluster under the exercises directory. You need that folder as it contains a .submit file that allows you to submit the project. Use the following command to copy the directory to your 216 directory:

```
cp -r ~/216public/exercises/draw_figures ~/216
```

It is important that you develop your code in the draw_figures directory you copied.

Before you read the specifications, you should check the example provided in Section 11, Sample Output. This will give you a clearer picture of what your program is intended to achieve. The set of explanatory prompt messages, error messages, and output messages that are presented in the example, are exactly the ones you are required to use in your program. For this assignment, you will write a program that draws rectangles and triangles.

Unlike other assignments for this course, you can work together with other classmates, but you may NOT exchange any code.

If you are on the waitlist for the course, you will not have access to grace. You can start working on this assignment by using online systems that compile C code. Just make sure your code is not accessible by others. Contact your lab TA for assistance.

3 Piazza

Make sure you are part of Piazza, otherwise you will not receive important information about class assignments. If you are not part of Piazza, e-mail one of your TAs.

4 Submit Server Time

This exercise is due at 11:30 pm. You will notice that in the submit server the due time is set to 11:31 pm. This is because a submission that is done exactly at 11:30 pm is considered late by the submit server. Make sure you do not wait until the last minute to submit.

Before you do anything else, make sure you can submit your project. Information on how to submit assignments can be found in Section 8, Submitting Your Assignment.

5 Regarding Which Submission Gets Graded

In this course the submission we grade is the one that has the highest score (only based on tests) in the submit server. Please keep this in mind for all class projects / exercises. You should not wait until the project is due to edit your code so it satisfies style requirements or any other requirements.

For example, for this exercise, 15 pts are associated with defininining functions. The submit server does not check for the functions, so you can pass all the submit server tests without them. The following scenarios are possible:

1. Submission that passes some of the tests (test total 50 pts), but does not satisfy the functions requirement.

2. Submission that passes some of the tests (test total 49 pts), and satisfies the functions requirement.

We will grade the first submission, instead of the second. For the first one you will receive a total score of 50 pts; if we were to grade the second, 64 pts. One reason why we grade the first one is that we want you to implement code (e.g., style and requirements from the beginning). Also, the submit server is limited in terms of the options we have regarding project grading.

6 Specifications

1. The program will read an integer value representing a request. The possible request values are:
   a. 0: Quit the program
   b. 1: Draw a rectangle given a character, width, and length.
   c. 2: Draw a triangle given a character and size.
   d. 3: Other figure.
   e. Any other value is considered invalid and the message "Invalid choice." will be printed.
2. The program will keep reading requests until 0 is provided.
3. The program will generate the message "Invalid data provided." if:
   a. 0 or a negative value is provided for width, length, or size.
   b. The only valid characters are *, %, and #.
4. The program will print the message "Bye Bye." before finishing.
5. Reading characters in C can be tricky as the newline character (enter) is considered a character. Use a space in the scanf format string to indicate that spaces must be skipped.

7 Project Requirements

1. Your grade will be based on the results obtained from the submit server. It is your responsibility to verify that your program generates the expected results in the submit server.
2. Your program must be named exactly draw_figures.c, otherwise it will not compile in the submit server.
3. All your C programs in this course should be written using the compiler `gcc`, with the options defined at
   http://www.cs.umd.edu/~nelson/classes/resources/setting_gcc_alias.shtml
4. Your program should be written using good programming style as defined at
   http://www.cs.umd.edu/~nelson/classes/resources/cstyleguide/
5. You must have a function called draw_rectangle that has width, length, and a character as parameters. The function prints a rectangle based on the provided information. The function will return 0 if any parameter is invalid (no figure printing will take place in this case). The function will return 1 if printing took place.
6. You must have one function called draw_triangle that has size and a character as parameters. The function prints a triangle based on the provided information. The function will return 0 if any parameter is invalid (no figure printing take place) and 1 otherwise.
7. You must have a function called valid_character that determines whether a character used while drawing a rectangle or triangle is valid or not.

8. Do not use arrays for this assignment.

9. In the grace cluster you can verify whether your code is generating the expected results by using input / output redirection and the diff command. Information about diff and input / output redirection can be found at

   http://www.cs.umd.edu/~nelson/classes/resources/cdebugging/diff/

10. By mistake you can create a huge file while using output redirection. For example, an infinite loop will cause such a file. We (instructors) get a notification when you are using a lot of disk space. Make sure that after using output redirection, you remove all the result files you have created. Removing this kind of files applies to all class programming assignments (projects and exercises).

11. You don't need to implement option 3. We provided this option in case you would like to draw your own figure. If you draw something nice/cool add the following comment at the top of your program:

    ```
    /* GRADER OPTION 3 IMPLEMENTED */
    ```

    This will allow us to identify students implementing option 3. You will not receive extra credit for implementing this option. Some examples of how to generate colored text in C can be found in the colors_in_c.c file.

8 Submitting Your Assignment

1. In the assignment directory (draw_figures) execute the command **submit**. After providing your directory id and password, your code (.c file) will be uploaded to the submit server. Notice that some additional files might be uploaded (don't worry about it).

2. Your assignment must be electronically submitted by the date and time above to avoid losing credit. See the course syllabus for details.

3. Feel free to add any other functions.

9 Grading Criteria

Your assignment grade will be determined according to the following weights:

Results of public tests	25%
Results of release tests	60%
Function Requirement	15%

10 Academic integrity statement

Please **carefully read** the academic honesty section of the course syllabus. **Any evidence** of impermissible cooperation on assignments, use of disallowed materials or resources, or unauthorized use of computer accounts, **will be submitted** to the Student Honor Council, which could result in an XF for the course, or suspension or expulsion from the University. Be sure you understand what you are and what you are not permitted to do in regards to academic integrity when it comes to assignments. These policies apply to all students, and the Student Honor Council does not consider lack of knowledge of the policies to be a defense for violating them. Full information is found in the course syllabus– please review it at this time.

11 Sample Output

A sample execution of the program you need to write can be found below. The % symbol represents the Unix prompt. Be sure to test your program against a variety of inputs, so you are sure it works in all circumstances!

```
% a.out
Enter 1(rectangle), 2(triangle), 3(other), 0(quit): 1
Enter character, width and length: * 3 4
****
****
****
Enter 1(rectangle), 2(triangle), 3(other), 0(quit): 2
Enter character and size: * 5
    *
   ***
  *****
 *******
*********
Enter 1(rectangle), 2(triangle), 3(other), 0(quit): 2
Enter character and size: # 4
   #
  ###
 #####
#######
Enter 1(rectangle), 2(triangle), 3(other), 0(quit): 1
Enter character, width and length: a 3 4
Invalid data provided.
Enter 1(rectangle), 2(triangle), 3(other), 0(quit): 1
Enter character, width and length: * 0 4
Invalid data provided.
Enter 1(rectangle), 2(triangle), 3(other), 0(quit): 0
Bye Bye.
%
```