

# SMALL PROGRAM 3

---

COP3223C Introduction to Programming with C  
Dr. Andrew Steinberg

Fall 2022

## Due Date

The assignment is due on September 24th 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 *smallprogram3\_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 *smallprogram3\_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!!

## The Python Script File! Read Carefully!

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. The script does not point directly where your mistake(s) are in the code. It will only produce a success or unsuccessful output message as a whole. If you get an unsuccessful output my suggestion is to look at the sample solution text file provided to see what is different from your answer and Dr. Steinberg's when comparing. If you have extra white space or new lines or even just missing a space/new line, you will lose points that won't be changed!

Make sure you place the python script in the same directory as your C file. You can use the ls command to check to see if the following items are in the directory.

---

```
ls
```

---

You should these three files in your directory.

1. Your C File.
2. The Python Script File
3. Sample Solution Text File

If you have these three files. You are ready to run the script. Use the following command to test your code with Dr. Steinberg's provided solution sample.

---

```
python3 sp3test.py
```

---

If the script says your output is incorrect, checkout the sample text file that was generated (a new text file will be created from the script that contains YOUR output). If your numbers are off or different from Dr. Steinberg's, then that means there is something not right with code's logic and calculating the answer. However if your numbers match with Dr. Steinberg's solution, then that means there is extra/missing white space or newlines detected. Compare the text file generated by the script with solution text file line by line to find the missing/extra white space or newlines. Once you believe you found the error, rerun the script to see if the output matches.

## 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 3
```

---

Missing a comment header will result in point deductions!

## The Solution Text File

You are provided a solution file that was created by Dr. Steinberg's Python Script. Now you may notice some strange things about the file. In this assignment, you are going to write statements that involve interacting with the user. Now you are probably wondering from the solution text file where the interaction is happening? The fact is that the Python script handles the interaction. The script creates an input stream that feeds it input. That is why you don't see the input directly in the text file. For example, the text file on the first line says Enter a key from the keyboard: Lower!. Here this looks like we should be typing input, however the python script has already fed it input. That is why you don't see the values except for the results. In each problem, a screenshot of the C file being executed manually without the Python script shows how it looks on a normal run. Carefully look at the output in the pictures provided. Note: The arrow symbolizes that the text wrapped onto the next line of the pdf file. In the text file itself it is actually one whole line.

---

samplesolutionsp3.txt

---

Enter a key from the keyboard: Lower!

Hi! Thank you for calling the Superhuman Law Division at GLK&H!

Our associates are currently working hard for super people like you.

Please listen carefully to the options of who you would like to speak to in regards

↪ to your situation.

Option 1: Fined for thousands of dollars worth of damage to the city you were trying

↪ to protect.

Option 2: Accidentally create a sentient robot who got the feels and tried to

↪ destroy the world.

Option 3: You are an Asgardian god who unintentionally leaves a giant burning

↪ imprint on private

property every time you visit Earth.

Option 4: You just gained new superhero strength that is not recognized by the

↪ department of

damage control, and they are chasing you down after you performed a good deed.

Option 5: Your secret identity was revealed by a notorious person and now your

↪ personal life

is no longer the same.

Option 6: Another super being issue that was not mentioned previously in the options

↪ given.

Selection: You have selected option 4. Do not worry! We will talk to the department

↪ of damage control.

Enter the x-coordinate: Enter the y-coordinate: 2.53, -4.29 is in quadrant IV.

Enter three sides separated by a whitespace: Checking these logistics from the

↪ input.

1 + 2 > 3

1 + 3 > 2

2 + 3 > 1

Not Triangle!

---

## Problem 1

Write the definition of a user-defined function called `letters`. The function will determine if a character value is a letter in the English Alphabet. The function has no parameters. Inside, the definition, you will prompt the user to input a character. If the letter is lowercase, then the message "Lower !" is displayed to the terminal. If the letter is uppercase, then the message "Upper !" is displayed to the terminal. If the character value is not part of the alphabet, then the message "Not a Letter !" is displayed to the terminal. *Hint, think about what happens when characters are evaluated.* The following figure shows a sample output scenario when the letter is lowercase. Make sure your output matches to receive potential credit. Do not worry if the user enters multiple characters at once. We have not learned strings yet!

```
Enter a key from the keyboard: h
Lower!
```

Figure 1: Sample output when the key entered from the keyboard is a lowercase letter from the English Alphabet.

## Problem 2

You are a super being in need of a super lawyer. You learned that Jennifer Walters is in your city representing super humans who may unintentionally leave a mess behind from saving the day. You dial the number 1-877-SHE-HULK. You connect with the operator who sounds like Jennifer Walters telling you all of the services that the Superhuman Law Division at GLK&H offers. Write

Option	Service Representation Needed in...	Response from option selection...
1	Fined for thousands of dollars worth of damage to the city you were trying to protect.	You have selected option 1. Do not worry! I will make sure you don't give up that green.
2	Accidentally create a sentient robot who got the feels and tried to destroy the world.	You have selected option 2. Do not worry! You probably had a good intention that just didn't work out properly.
3	You are an Asgardian god who unintentionally leaves a giant burning imprint on private property every time you visit Earth.	You have selected option 3. Do not worry! I understand you like to make a grand entrance.
4	You just gained new superhero strength that is not recognized by the department of damage control, and they are chasing you down after you performed a good deed.	You have selected option 4. Do not worry! We will talk to the department of damage control.
5	Your secret identity was revealed by a notorious person and now your personal life is no longer the same.	You have selected option 5. Do not worry! We will make sure that you are protected. Please be cautious if you decide to seek Dr. Strange's help.
6	Another super being issue that was not mentioned previously in the options given.	You have selected option 6. Do not worry! I will make sure to use my superhero mind to solve your super problem!

Figure 2: Table for problem 2 that shows the message for each option.

a user defined function called `greenLawyer` that displays the list of options, allows the user

```

Hi! Thank you for calling the Superhuman Law Division at GLK&H!
Our associates are currently working hard for super people like you.
Please listen carefully to the options of who you would like to speak to in regards to your situation.
Option 1: Fined for thousands of dollars worth of damage to the city you were trying to protect.
Option 2: Accidentally create a sentient robot who got the feels and tried to destroy the world.
Option 3: You are an Asgardian god who unintentionally leaves a giant burning imprint on private
property every time you visit Earth.
Option 4: You just gained new superhero strength that is not recognized by the department of
damage control, and they are chasing you down after you performed a good deed.
Option 5: Your secret identity was revealed by a notorious person and now your personal life
is no longer the same.
Option 6: Another super being issue that was not mentioned previously in the options given.
Selection: 4
You have selected option 4. Do not worry! We will talk to the department of damage control.

```

Figure 3: Sample output for problem 2. This shows the welcoming message along with the result of the option selected.

to select one, and then display the result. If none of the values match, then the message "I'm sorry. I don't recognize that super being option." is displayed to the terminal window. Inside the user defined function you will display a welcoming message and list the options for the user. Figure 3 shows a sample text to be displayed as the welcoming message along with selecting the output and the result. Make sure it is exact to receive potential credit as the script will check for it. **You must also use a switch statement for this question to receive full points. If a switch statement is not used then the highest mark you can receive is half mark on the rubric evaluation.**

### Problem 3

Write a user defined function definition called `coordinates` that takes x-y coordinates of a point in the Cartesian plane and prints a message telling either an axis on which the point lies or the quadrant in which it is found.

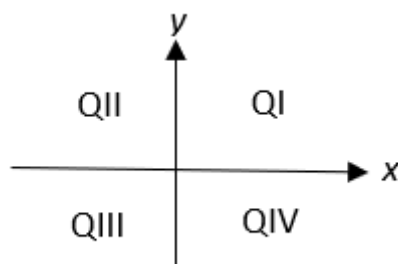


Figure 4: Quadrants in the 2D coordinate plane.

The user defined function takes two double arguments and doesn't return anything. *Hint: Don't forget x-axis, y-axis and origin.* The coordinates are collected in the main function. The following figure shows a sample output. Make sure the output matches for the script to receive credit. The values of the coordinates must be displayed up to two decimal places.

```
Enter the x-coordinate: 2.53
Enter the y-coordinate: -4.29
(2.53, -4.29) is in quadrant IV.
```

Figure 5: Sample output for problem 3. Make sure your output is in this format for the script!

## Problem 4

You are working for a architecture company that loves designing triangle shape buildings. The company has grown that they are able to create their own software to perform some blue print designs rather than having someone design manually by hand. Part of the software is configuring measurements to ensure that the potential building is in the shape of a valid triangle. Figure 7 shows the geometric definition of what determines a shape to be a valid triangle. Using

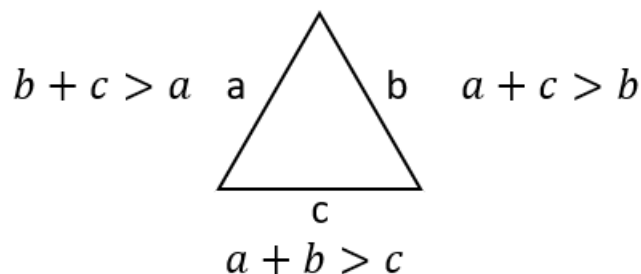


Figure 6: The triangle and its properties of validation.

these properties, write the definition of a user-defined function called `triangle`. The function has three parameters that are all integer type. Each parameter represents the length of the side. Assume that the input is always collected in abc order. Inside the main function, you will ask the user for the length of each of the sides. Once the input is collected, you will call the user-defined function and perform the respective operation that was asked in this problem. The function will display the logistics it is checking to determine if the values represent a triangle or not. The function also returns an integer value which represents the outcome. The value 1 returned means the shape is a valid triangle. The value -1 means otherwise. Using those values, you will then display the outcome message in the main function. If the value is 1, then "Triangle!" is displayed. Otherwise display "Not Triangle!". The following figure shows a sample output of what the python script expects. Make sure to have it exact in terms of white space, newlines, and characters or else the script will mark it wrong.

```
Enter three sides separated by a whitespace: 1 2 3
Checking these logistics from the input.
1 + 2 > 3
1 + 3 > 2
2 + 3 > 1
Not Triangle!
```

Figure 7: Sample output of problem 4. Make sure your output is in this format for the script!