CS 118 — Programming Fundamentals
Assignment #5

Due Date: Friday, October 11th at 11:55pm on Google Classroom.

Instructions: Assignments are to be done individually. **No late assignments will be accepted.** You must complete this assignment by yourself. You cannot work with anyone else in the class or with someone outside of the class. The code your write must be your own. You are encouraged to get help from the instructional staff. You may post general questions on Piazza. Do not post more than one line of code when using Piazza.

You must **submit a single zip file** containing your code and documentation on Google Classroom named ⟨*your_student_id*⟩*.zip* where ⟨*your_student_id*⟩ is something like *i19-XXXX*. This means that you must submit only **one file named *i19-XXXX.zip* containing only your source files**. Each file that you submit **must contain your name, student-id, and assignment#** on top of the file in comments. You submission must NOT contain multiple main() functions, otherwise it will not compile for grading. Test your program on a lab machine before submission.

Follow the instructions. Assignments not following the instructions will be awarded zero points.

Assignment Statement: ASCII Art is a graphics design technique that uses keyboard characters to create drawings and images. Recall Assignment 2 in which you printed the ASCII art representation of the FAST Main Building. In this assignment you will write a program that produces the ASCII art representation of the FAST Main Building where the user can give the dimensions of the building as input.

The output will vary based on two inputs that the user will provide; *number of floors* and *width* of the bulding measured by the number of windows in the top floors. By simply changing the input, the output will change (somewhat) dramatically. In the end your program must match the outputs shown for the given dimensions EXACTLY. The window and grass style are fixed for this assignment.

Here is the program output with the with inputs 4 and 13 for *number of floors* and *width*, (also provided as out1.txt).

```
FAST Main Building in ASCII Art
Enter number of floors: 4
Enter width of building: 13
                          !
                        (***)
                       |O|O|O|
   /\_________________---------_________________/\
  | || O O O O || O || O O O || O || O O O O || |
  ==================== _____ ====================
  | || O O O O || O || |   | || O || O O O O || |
  | || O O O O || O || |   | || O || O O O O || |
  | || O O O O || O || |___| || O || O O O O || |
 /                  /  ___  \                  \
/ * * * * * * * * * /  /___\  \ * * * * * * * * * \
```

Your program should take in both user inputs using two `scanf()` statements as shown in the sample. The inputs cannot be empty or newlines. Our assumption is that they will be positive integers.
When the input values are changed to 7 and 19, the following output is produced:

```
FAST Main Building in ASCII Art
Enter number of floors: 7
Enter width of building: 19
                                !
                               (***)
                              |O|O|O|
   /\_____________________---------_____________________/\
  | || O O O O O O O || O || O O O || O || O O O O O O O || |
  | || O O O O O O O || O || O O O || O || O O O O O O O || |
  | || O O O O O O O || O || O O O || O || O O O O O O O || |
  | || O O O O O O O || O || O O O || O || O O O O O O O || |
  ========================= _____ =========================
  | || O O O O O O O || O || |   | || O || O O O O O O O || |
  | || O O O O O O O || O || |   | || O || O O O O O O O || |
  | || O O O O O O O || O || |___| || O || O O O O O O O || |
 /                          /  ___  \                        \
/ * * * * * * * * * * * * /  /___\  \ * * * * * * * * * * * * \
```

The width of the building will be an odd number. If the user enters an even number, the closest odd number smaller than the user input will be chosen as the width. So, 7, 19 and 7, 20 will produce the same output given above. The minimum width of the building is 3. If the user enters a width less than 3, the base structure is printed out as follows.

```
FAST Main Building in ASCII Art
Enter number of floors: 4
Enter width of building: 2
       _____
       |   |
       |   |
  ____ |___| ____
 /   /  ___  \   \
/ * /  /___\  \ * \
```

This assignment is meant to give you practice with problem decomposition, loop statements, nested loops and conditionals. You may use all the statements and constructs that you have learnt unilt now.

The number of floors in the building design can also vary. The designers at FAST considered various heights of the building. Initially a 2 storey building was constructed without a dome as can be seen in an earlier image of FAST. So any construction of fewer than 3 floors would not have a dome on it. It was also decided that if the building rises to eight or more floors it

Figure 1: FAST 2 Storey.

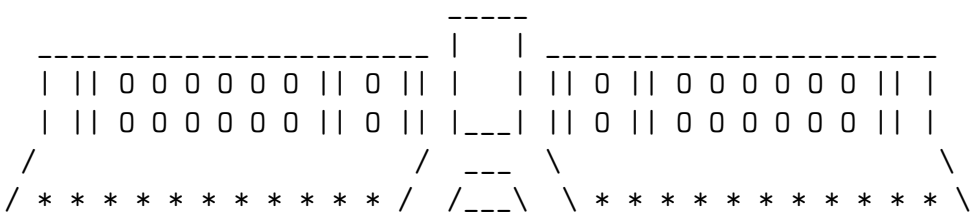
would feature two water tanks on the top corners of the building (just like we have in the new building!). We will stick to the same plan!

This means that if the user inputs eight or more floors, two tanks will be printed instead of the dome structure as seen below.

```
FAST Main Building in ASCII Art
Enter number of floors: 8
Enter width of building: 10
  ______                          ______
  |    |                          |    |
  |    |__________________________|    |
  | || O O || O || O O O || O || O O || |
  | || O O || O || O O O || O || O O || |
  | || O O || O || O O O || O || O O || |
  | || O O || O || O O O || O || O O || |
  | || O O || O || O O O || O || O O || |
  ================ _____ ================
  | || O O || O || |   | || O || O O || |
  | || O O || O || |   | || O || O O || |
  | || O O || O || |___| || O || O O || |
 /              /  ___  \              \
/ * * * * * * * /  /___\  \ * * * * * * * \
```

However, if the height of the building is less than three floors, it will not feature any of these structures as shown below.

```
FAST Main Building in ASCII Art
Enter number of floors: 2
Enter width of building: 17
```



You should use functions to structure your solution. Try not to write your complete solution in `main()`.

You are required to properly indent your source code and will lose points if your indenting is not readable and consistent. You should localize variables whenever possible. Include a comment for each function you write explaining the purpose of that method.

Note that the window and grass styles are fixed and the user inputs (number of floors, width) will determine how the drawing is rendered. The inputs to your program will always be two positive integers.

For your reference the executable *build* has been provided for purposes of comparison. This file may not execute on MAC and virtual machines. You can run this file using the commands:

```
chmod 744 build
./build
```

On any given execution your program will produce just one version of the building, but it should be possible to change the height and width during different runs. Given are a few output files for various inputs. Your output must match these exactly for the given inputs or you will lose points for correctness. Use a `diff` tool to ensure your program produces the correct output.

Style issues. We will grade program hygiene as well as correctness. Did you provide a good structure to the program using functions? Did you minimize the scope of variables to the smallest necessary? Did you use meaningful identifiers? Did you provide consistent tabbing and spacing for code inside functions and if statements? Did you provide comments for your functions?

Honor Policy

This assignment is a individual learning opportunity that will be evaluated based on your ability to think independently, work through a problem in a logical manner solve the problems on your own. You may however discuss verbally or via email the general nature of the conceptual problem to be solved with your classmates or the course instructor, but you are to complete the actual assignment without resorting to help from any other person or other resources that are not authorized as part of this course. If in doubt, ask the course instructor. You may not use the Internet to search for solutions to the problem.