

# Classwork 7: Bar Graphs

In-class Date: Wednesday 10 April

Due Date: Wednesday 17 April

## Objectives

To gain more practice with nested `while` loops.

## Assignment

For this assignment, you will write a program which prints horizontal “bar graphs” using numbers entered by the user.

### Example Compilation and Execution

```
[arsenaul@linux1 cw7]$ gcc -Wall bar.c
[arsenaul@linux1 cw7]$ ./a.out
Enter a positive number (type 'quit' to end): 3
***
Enter a positive number (type 'quit' to end): 17
*****
Enter a positive number (type 'quit' to end): 23
*****
Enter a positive number (type 'quit' to end): 22
*****
Enter a positive number (type 'quit' to end): 9
*****
Enter a positive number (type 'quit' to end): 4
****
Enter a positive number (type 'quit' to end): quit
[arsenaul@linux1 cw7]$ ./a.out
Enter a positive number (type 'quit' to end): quit
No numbers were entered
[arsenaul@linux1 cw7]$
```

The “bar” in this bar graph are the asterisks (\*) printed after each user entry. The number of asterisks must be equal to the number entered by the user.

### Starter Code

Use this code to help you get started.

```
/*****
** File:    bar.c
** Author:  <myName>
** Date:    <todaysDate>
** Section: CMSC104 Section 02
```

```

** E-mail:    <myEmailAddress>
**
** This file contains the main program for Classwork 7.
** TODO: <Explain what the assignment is asking you to do here.>
*****/

#include <stdio.h>

int main() {
    // Variable declarations
    int num;                /* user's positive number entry */
    int count = 0;          /* how many positive numbers user has entered */
    int stop = 0;           /* don't stop */
    int matched;            /* used like flag to see if user entered 'quit' */
    int innerLoopCounter;    /* how many iterations to print * */

    // TODO: Create outer loop for user input checking

    // TODO: Prompt user for positive number

    // TODO: If user typed 'quit', set stop to 1 so we can break out of outer loop;
    // else if user entered positive number, create inner loop for printing and
    //         updating count;
    // else notify the user they entered an invalid number

    // TODO: Let user know if no positive numbers were entered

    return 0;
}

```

## Notes

- Remember: Update the header comments with your information.
- You MUST use nested while loops for this assignment. The outer while loop should iterate once for each number entered by the user. The inner loop should print out the right number of asterisks.
- To print the asterisks, use a new variable and a while loop that counts up to the number entered by the user. (Each iteration of this inner while loop adds one to the counter.) For each iteration of the inner while loop, print out a single asterisk without a newline. After the loop ends, your program should print out a newline character. This way, the asterisks appear on one line.
- Use “min.c” on Blackboard as an optional reference which shows how the outer loop could be written.
- Homework 7 will be a continuation of this assignment. If you are done early, you can continue

to Homework 7 (when posted), but be sure to submit the appropriate files to the correct assignment!

## Grading Rubric

- Header comment: 2 points
- Body comments: 3 points
- Compiles: 40 points
- Displays “No numbers entered” appropriately: 10 points
- Two while loops: 25 points
- Correct number of asterisks printed: 20 points

## What to Submit

Use the `script` command to record yourself compiling and running your program several times with different numbers. (Do not record yourself editing your program!) Exit from script, submit your program and the typescript file.

```
[arsenaul@linux1 cw7]$ submit cmsc104_arsenaul cw7 bar.c typescript
```

## Verify

Make sure you submitted the assignment correctly.

```
[arsenaul@linux1 cw7]$ submitls cmsc104_arsenaul cw7
```