

Fall 2009

Introduction to C - Programming Assignment #1

Due Date: September 9, 2009 – 11:59pm

Objective

1. To give students practice at typing in, compiling and running a simple program.
2. To learn how to read in input from the user.
3. To learn how to use assignment statements and arithmetic expressions to make calculations.

Problem A: Buying UCF Football Tickets

You are excited to be in your first semester at UCF but are disappointed that football tickets are allocated through a lottery. Since you had a successful business in high school, you have plenty of money to buy your own football tickets so that you can guarantee your seats. In order to simplify the process, you decide to use your newly acquired C programming skills to aid you (and anyone else) in calculating the total cost of purchasing some football tickets.

Here is the information your program will ask the user to enter:

1. The number of lower bowl tickets he/she wants to buy.
2. The number of upper bowl tickets he/she wants to buy.
3. The number of games worth of tickets he/she wants to buy.
4. The sales tax in the locale in which the tickets are being bought, as a percentage.

Prompt the user to input these four values. You will output a single statement with the total cost of the football tickets.

Please define the following constants in your code:

```
#define LOWER_BOWL_TIX_PRICE 50.00
#define UPPER_BOWL_TIX_PRICE 25.00
```

Input Specification

1. The number of lower and upper bowl tickets will be integers in between 0 and 100, inclusive.
2. The number of games the user will be buying tickets for will be in between 0 and 7, inclusive.
3. The sales tax will be a real number percentage in between 0% and 20%.

Output Specification

Output the cost for all of the football tickets with tax rounded to two decimal places. Your output should follow the format below, where XX.XX is the cost in dollars for all of the football tickets.

The total cost of your football tickets is \$XX.XX.

Output Sample

Here is one sample output of running the program. Note that this test is NOT a comprehensive test. You should test your program with different data than is shown here based on the specifications given. The user input is given in *italics* while the program output is in bold.

Sample Run

How many lower bowl seats do you want?

2

How many upper bowl seats do you want?

4

For many games do you want to buy these tickets?

5

What is the sales tax percentage in your locale?

6.5

The total cost of your football tickets is \$1065.00.

Problem B: UCF Parking Woes

Now that UCF has over 52,000 students, even with a parking permit, actually obtaining a parking spot is more difficult than any computer science test you'll have to take. Unfortunately, there is nothing you can do to alleviate your parking woes. But, using your newly found C programming skills, you can at least write a program that can predict how long it will take you to find a parking spot, which may aid you in planning how early you need to come to campus.

Prompt the user to enter the following values:

1. Time of Day (hour from 0-23, in military time, *let this be t*)
2. Parking Spots Available (for the semester, *let this be a*)
3. Parking Permits Given (for the semester, *let this be g*)

After much observation, you have realized that the formula for determining the amount of time you will wait to get a parking spot is:

$$\text{Time (in minutes)} = (12 - |12 - t|) * g / a$$

Input Specification

1. The time of day will be a single integer in between 0 and 23, inclusive, representing the hour the user is attempting to park.
2. This number of spots available will be a positive integer less than 50,000.
3. The number of permits given will be a positive integer less than 60,000.

Output Specification

Output the waiting time for parking in total number of minutes (rounded to the nearest hundredth) with the format below (XX.XX is the time in minutes):

You will have to wait XX.XX minutes to find parking.

Output Sample

Here is one sample output of running the program. Note that this test is NOT a comprehensive test. You should test your program with different data than is shown here based on the specifications given. The user input is given in *italics* while the program output is in bold.

Sample Run

What hour are you looking for parking?

10

How many spots are available this semester?

9000

How many parking permits were given this semester?

45000

You will have to wait 50.00 minutes to find parking.

Problem C: UCF Meal Plan

Your parents have gotten you a flex card meal plan, where you can choose to eat at many fine dining establishments at or near campus. Although you have many, many options of where to eat, you have quickly narrowed down a simple strategy:

You will ALWAYS eat at either Subway or Lazy Moon.

You know you want to avoid the freshman fifteen, so you've included Subway on your list. Of course, it helps that Subway is relatively inexpensive.

But, like all college students, you love pizza, and Lazy Moon undoubtedly has some good pizza. Unfortunately, what you like to buy at Lazy Moon costs more than what you like to buy at Subway.

Thus, after determining how many times you will have to eat out the whole semester, your goal is to maximize the amount of times you eat at Lazy Moon without going over the allotted amount of money your parents have given you to spend on your flex card. Furthermore, you'd like to know how much money, if any, will be left on your card after the semester is over.

You will prompt the user for the following information:

1. The cost of a meal at Subway (in dollars – must be an integer).
2. The cost of a meal at Lazy Moon (in dollars – must be an integer and greater than #1).
3. The number of meals you will need to eat for the semester.
4. The amount of money (in dollars – must be an integer) your parents have put on your meal card.

Input Specification

1. The cost of both meals will be positive integers less than 20 with the cost of a meal at Lazy Moon being more expensive than the cost of a meal at Subway.
2. The number of meals will be a positive integer less than 300.

3. The amount of money your parents place on your meal card will be a positive integer that is somewhere in between the cost of eating every meal at Subway and the cost of eating every meal at Lazy Moon, inclusive.

Output Specification

Output the number of meals you will eat at both Subway and Lazy Moon, as well as the amount of money leftover on your card at the end of the semester. Follow the format shown below:

You will eat X meals at Subway and Y meals at Lazy Moon.
You will have \$Z left on your card.

X, Y and Z represent the appropriate integers.

Output Sample

Here is one sample output of running the program. Note that this test is NOT a comprehensive test. You should test your program with different data than is shown here based on the specifications given. The user input is given in *italics* while the program output is in bold.

Sample Run

How much do you spend on a meal at Subway?

4

How much do you spend on a meal at Lazy Moon?

6

How many meals will you eat this semester?

150

How much money did your parents put on your meal card?

749

You will eat 76 meals at Subway and 74 meals at Lazy Moon.

You will have \$1 left on your card.