Section 2: Introduction to C - Programming Assignment #1

Due date: September 7, 2023

Notes

1. Please read the notes on WebCourses Submission provided with AssignmentZero if you didn't get a chance to do that Assignment.

Objectives

1. To give students practice at typing in, compiling and running simple programs.
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
4. To learn how to use the if-then-else construct (only Problems B and C, not A).
5. To learn how to use the for-loop construct (Problem C).

Introduction: UCF's Theme Park

Now that UCF is one of the largest universities in the nation (in terms of number of undergraduates enrolled), the decision has been made for UCF to pursue bigger goals. Since Orlando is the capital of the tourism industry, UCF has decided to enter the theme park market, to compete with Disney and Universal. Of course, UCF has good knowledge and is intelligent, so the plans will include appropriate Covid practices: everything will be socially distanced, and strong masks will be mandatory. After hearing about your exquisite programming skills from your Intro to C Programming instructor, UCF has decided to hire you to write some of the software necessary to help the theme park UCF is designing, Universally Comical Funland. In the following assignment you will write three separate programs to be used in the theme park!

Problem A: Gift Shop Purchase (shop.c) (15 points)

Everyone knows that the first component every theme park needs is a gift shop to aid in revenue. Each of the many gift shops at Universally Comical Funland (UCF) will be equipped with the same type of cash register that needs a small program to make calculations for tourist purchases. For this problem, you'll design a program that completes a simple type of transaction.

You will ask the user for the price of the item they are buying, the quantity of the item they are buying, and whether or not the item is a taxed item. The user's input for the last question will be 0 if the item is NOT taxed, and 1 if it is taxed. The tax rate for UCF should be defined in your program as a constant as follows:

```
#define TAX_RATE 0.065
```

Note: The intention is to solve this problem WITHOUT an If statement or a Switch statement. Full credit will only be given to solutions that avoid an If and Switch. Also, do not use a For-loop or a While-Loop.

Input Specification
1. The item price will be a positive real number less than 100.
2. The quantity of the item purchased will be a positive integer less than 100.
3. The answer to the tax question will be an int, it will either be 0 (no tax), or 1 (tax).

Output Specification
Output the total cost of the purchase (in dollars) to two decimal places using the format below:

```
Your total purchase will cost $X.XX.
```

Note: the number of digits before the decimal will vary, based on the cost of the purchase, but for dollar amounts, you should always print exactly 2 digits after the decimal.

Output Sample
Below is one sample output of running the program. **Note that this sample is NOT a comprehensive test.** You should test your program with different data than is shown here based on the specifications given above. In the sample run below, for clarity and ease of reading, the user input is given in *italics* while the program output is in **bold**. (Note: When you actually run your program no bold or italics should appear at all. These are simply used in this description for clarity's sake.)

Sample Run #1
What is the cost of the item to be purchased (in dollars)?
9.75
How many of the item are you purchasing?
10
Is the item a taxed item (1 = yes, 0 = no)?
1
Your total purchase will cost $103.84.

(Note: Had the user entered 0 for the last question in this example instead, the cost would have been $97.50.)

Problem B: Calorie Counter (calories.c) (20 points)
In order to be successful, UCF needs to provide its tourists with helpful features that Disney and Universal don't have. Since people are more health conscious these days, UCF has decided that each park visitor will receive a calorimeter, which will gauge his/her net calorie gain or loss during the day.

The secret of course is that this device isn't perfect. It doesn't ACTUALLY measure the number of calories you burn or the number of calories you ingest. Instead, it makes four assumptions:

1) When walking, you burn 5 calories/minute.

2) When standing in line, you burn 2 calories/minute.
3) When you are consuming solid food, you ingest 40 calories/minute.
4) When you are consuming a drink, you ingest 20 calories/minute.

The calorimeter is "smart enough" to know which of the four activities you are engaging in at any point in time, and at the end of the day, compiles a tally of how many minutes you spent doing each of these four activities. Write a program that takes these four values as input and outputs the pounds lost (we want a positive message for our clients). If the user has gained weight, print out a negative number for weight lost. Note, use the following constants for this problem:

```
#define WALK_CAL 5
#define STAND_CAL 2
#define DRINK_CAL 20
#define FOOD_CAL 40
#define CALS_PER_POUND 3500
```

Input Specification
1. All four values will be non-negative integers less than 720, representing the number of minutes spent for each of the four activities.

Output Specification
When weight is lost, output a single line of the following format telling the user how many pounds they lost, rounded to 3 decimal places.

```
You lost X.XXX pounds today!
```

If weight is gained, output a single line of the following format, print out the weight change as a negative number rounded to 3 decimal places.

```
Weight lost is -X.XXX pounds.
```

Output Samples
Samples of the program running are included below. **Note that these samples are NOT a comprehensive test.** You should test your program with different data than is shown here based on the specifications given above.

Sample Run 1

```
How many minutes were you walking?
120
How many minutes were you standing?
230
How many minutes were you drinking?
20
How many minutes were you eating?
15
```

```
You lost 0.017 pounds today!
```

Sample Run 2

```
How many minutes were you walking?
120
How many minutes were you standing?
230
How many minutes were you drinking?
20
How many minutes were you eating?
17
Weight lost is -0.006 pounds.
```

Problem C: Roller Coaster Design (coaster.c) (65 points)

Every great theme park has a signature roller coaster. In designing a roller coaster, we must decide how many trains of cars need to be placed on the track. It turns out that no more than 25% of the track can be occupied with these trains to adhere to safety regulations. Thus, if the track is 1000 feet long, and a train is 42 feet long, then up to 5 trains can fit on the track. (To see this, note that 5 trains have a total length of 210 feet, and this value is 21%, less than or equal to 25%, of the total track length of 1000 feet. Note that 6 trains would take up 252 feet, or 25.2% of the track, which is too much.)

In a train of cars, the first car is 10 feet long and all subsequent cars are 8 feet long. All cars seat up to 4 people. Since each of these values is constant, please use the following constants to store them:

```
#define FIRST_CAR_LENGTH 10
#define NORMAL_CAR_LENGTH 8
#define CAR_CAPACITY 4
```

For this program, the user will enter the total length of the track and the maximum length of the trains for the track. It is assumed that the trains formed will be as long as possible. For example, if the user enters 30 for the maximum length of the train, then the actual trains will have three cars and be of length 26, since a four car train would exceed 30 feet. Your program should calculate the number of people that can be supported on the track at one time. (Note: It may be the case that more people can be supported by making a shorter train than possible, but for this particular assignment, maximize the size of each train.)

Input Specification

1. The total length of the track will be a positive integer (in feet) less than 10000.
2. The maximum length of a train will be a positive integer in between 10 and 100.

Output Specification
Output the maximum number of passengers on the roller coaster at any one time with a single statement of the following format:

```
Your ride can have at most X people on it at one time.
```

Early Output Samples (later, Final outputs must look like Sample Runs 3 and 4)
At first, two sample outputs of running the program are included below. **Note that these samples are NOT a comprehensive test.** You should test your program with different data than is shown here based on the specifications given above.

Sample Run #1
```
What is the total length of the track, in feet?
1000
What is the maximum length of a train, in feet?
42
Your ride can have at most 100 people on it at one time.
```

(Note: Each train has 5 cars on it, and each car has 4 people, at most. Thus, 20 people can sit in one train. As previously established, 5 trains is the maximum for this track, so the maximum number of people this roller coaster can support is 100.)

Sample Run #2
```
What is the total length of the track, in feet?
1000
What is the maximum length of a train, in feet?
49
Your ride can have at most 100 people on it at one time.
```

Now, modify what you have written to do:
If the Maximum Length exceeds the Actual Length print a message that "Maximum Train Length has surplus of %d feet" (and if no excess, print "Maximum Length fits exactly"); secondly, embed all of this (part C) code in a FOR loop that runs N times, where N is to be (prompted for and) read in at the beginning of the program. Thus, the user will input N and then will input the track_length and train_maximum N number of times.
Given these new requirements, see Sample Runs #3 and #4. Your program must read the new input for the value of N, and **must generate output similar to Sample Runs #3 and #4.**

Sample Run #3

```
What is the value for N?
3
```

```
What is the total length of the track, in feet?
1000
What is the maximum length of a train, in feet?
49
Your ride can have at most 100 people on it at one time.
Maximum Train Length has surplus of 7 feet


What is the total length of the track, in feet?
1000
What is the maximum length of a train, in feet?
42
Your ride can have at most 100 people on it at one time.
Maximum Length fits exactly


What is the total length of the track, in feet?
1000
What is the maximum length of a train, in feet?
18
Your ride can have at most 104 people on it at one time.
Maximum Length fits exactly
```

Sample Run #4

```
What is the value for N?
1

What is the total length of the track, in feet?
1000
What is the maximum length of a train, in feet?
59
Your ride can have at most 112 people on it at one time.
Maximum Train Length has surplus of 1 feet
```

Deliverables

Three source files:

1) *shop.c*, for your solution to problem A
2) *calories.c* for your solution to problem B
3) *coaster.c* for your solution to problem C

All files are to be submitted over WebCourses.

Restrictions

Although you may use other compilers, your program must compile and run using Codeblocks. Each of your three programs should include a header comment with the following information: your name, course number, section number, assignment title, and date. Also, make sure you include comments throughout your code describing the major steps in solving the problem.

Grading Details

Your programs will be graded upon the following criteria:

1) Your correctness

2) Your programming style and use of white space. Even if you have a plan and your program works perfectly, if your programming style is very poor or your use of white space is very poor, you could get 2% deducted from your grade.

3) Compatibility to Codeblocks. If your program does not compile in this environment, you will get a **sizable** deduction from your grade.