Fall 2016 COP 3223H Program #4: Ads, flags and debates!
Due date: Please consult WebCourses for your section

Objectives

1. To learn C syntax to solve problems that utilize assignment statements.
2. To learn C syntax to solve problems that utilize if statements.
3. To learn C syntax to solve problems that utilize loops.

Problem A: Advertising (ads.c)

In order to gain votes, the candidates buy various advertisements in the media to reach voters. Three of the most common forms of advertisement are: (a) TV, (b) Radio, (c) Online. The cost model for all three vary.

Commercial time is based on a straight per second rate. If a candidate's commercials play for 240 seconds and the fee is $10/second, then $2400 is the total cost of the commercials.

Radio charges a flat fee of $10000 for introducing a new commerical and then charges a much smaller fee of 50 cents per 30 second commercial.

Online ads cost is proportional to the number of times the ads are clicked. Each click costs one penny.

Write a program that askes the user to enter the number of seconds of commerical time being bought, the number of 30 second radio commericals being bought and the number of times an online ad was clicked and determines the total cost for advertising.

Input Specification

The number of seconds of commerical time being bought will be a positive integer less than 1,000,000. The number of radio commercials bought will be a positive integer less than 100,000. The number of clicks for online ads will be a positive integer less than 1,000,000,000.

Output Specification

Output a single line with the following format:

```
The candidate's total advertising cost is $X.
```

where X is a dollar amount expressed to exactly two decimal places.

Sample Program Run #1

```
How much TV commercial time in seconds is being bought?
10000
How many radio commercials will be bought?
1000
How many clicks will the online ads receive?
987654
The candidate's total advertising cost is $21376.54.
```

Problem B: Am I out of time? (timelimit.c)

The debates have shown that neither of the candidates is very good at telling time, as they frequently go over the allotted time to speak during debates. Write a program to determine whether or not a candidate has gone over his/her allotted speaking time.

Your program will ask the user to enter the time limit for the segment of the debate in seconds. Next, your program will ask the user to enter the number of topics the candidate will speak on for the segment. Then, your program will ask the user for the number of seconds the candidate will speak for each topic.

Based on this, your program should determine if the candidate finished early, exactly on time, or late.

Input Specification

The first input value, *s*, the allotted number of seconds to speak will be a positive integer less than or equal to 1000. The number of topics the candidate will speak on, *t*, will be a positive integer less than or equal to 5. The amount of time the candidate will speak on each topic will each be positive integers less than or equal to 1000.

Output Specification

Based on the amount of time the candidate spent speaking, output a message using one of the three following formats:

```
The candidate finished X second(s) early!
The candidate finished right on time!
The candidate used X second(s) too many :(
```

where X represents the appropriate positive number of seconds.

Sample Program Run #1

```
What is the time limit in seconds for this segment?
300
How many topics will the candidate address?
3
How many seconds long will he/she speak on topic 1?
100
How many seconds long will he/she speak on topic 2?
99
How many seconds long will he/she speak on topic 3?
100
The candidate finished 1 second(s) early!
```

Sample Program Run #2

```
What is the time limit in seconds for this segment?
100
How many topics will the candidate address?
4
```

```
How many seconds long will he/she speak on topic 1?
25
How many seconds long will he/she speak on topic 2?
40
How many seconds long will he/she speak on topic 3?
15
How many seconds long will he/she speak on topic 4?
20
The candidate finished right on time!
```

Sample Program Run #3

```
What is the time limit in seconds for this segment?
500
How many topics will the candidate address?
1
How many seconds long will he/she speak on topic 1?
600
The candidate used 100 second(s) too many :(
```

Problem C: American Flag? (flag.c)

For your first exam, you designed a program that printed out a flag design similar to the stars of the American flag, though more generally. Implement the same program in C.

Input Specification

The number of rows will be a positive integer in between 1 and 20, inclusive. The number of stars on the first row will be a positive integer in between 2 and 40, inclusive.

Output Specification

Output the appropriate flag design, using the * character for each start.

Sample Program Run #1

```
How many rows in your flag design?
9
How many stars are on the first row?
6
* * * * * *
 * * * * * 
* * * * * *
 * * * * * 
* * * * * *
 * * * * * 
* * * * * *
 * * * * * 
* * * * * *
```

Sample Program Run #2

```
How many rows in your flag design?
4
How many stars are on the first row?
10
* * * * * * * * * * 
 * * * * * * * * * *
* * * * * * * * * * 
 * * * * * * * * * *
```

Deliverables

three source files:

1) *ads.c*, for your solution to problem A
2) *timelimit.c* for your solution to problem B
3) *flag.c* for your solution to problem C

All files are to be submitted over WebCourses.

Restrictions

Although you may use other compilers and coding environments, your program must run in Code::Blocks using a gcc compiler.

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 poor or your use of white space is poor, you could get 10% or 15% deducted from your grade.

Note: As mentioned in class, the input specifications are there to help you. Those are the requirements I guarantee I'll stick to when testing your program. You don't need to check if the user enters values within those parameters.