Fall 2009
Introduction to C - Programming Assignment #2

Due Date: Wednesday, September 23, 2009, 11:59pm

Objective

1. How to write and use an if statement.
2. How to write a single loop.
3. How to utilize compiler messages and printf statements for debugging.

Problem A: Parking Permits

Your latest triumph was writing a C program that calculated wait times for parking. The parking services people were so impressed with your program that they now want to hire you to write other programs for them. The first program they want from you deals with looking up the price of parking permits, based on the person getting the permit (student, staff, faculty) and the type of permit (sticker, hang tag). Your program will ask the user what type of person they are as well as what type of permit they would like. From this information, you must tell the user how much they must pay for their permit. Here is a chart with the necessary information:

Role Permit Type	Sticker	Hang Tag
Student	$100	$150
Staff	$200	$300
Faculty	$300	$450

Note: These are not the actual prices.

Input Specification

1. The first integer entered will be 1, 2 or 3, corresponding to student, staff or faculty, respectively.
2. The second integer entered will be 1 or 2, corresponding to a sticker or a hang tag, respectively.

Output Specification

Output the price of the permit for the user with a statement of the following format:

```
You need to pay $XXX for your parking permit.
```

where XXX represents the price of the user's parking permit in dollars.

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 role are you? (1=student, 2=staff, 3=faculty)
2
What type of tag do you want? (1=sticker, 2= hang tag)
1
You need to pay $200 for your parking permit.
```

Problem B: Dinner Schedule

You are excited that you've figured out how often you can eat at Lazy Moon, based on your budget. Now, because you are such a good student and planner, you'd like to lay out in advance which days you will eat dinner at Lazy Moon and which days you'll eat dinner at Subway.

As it turns out you have discovered that the following set of rules will work for your budget:

1) On all even days, you eat at Subway.
2) On all odd days that are also divisible by 5 **but NOT 25** (5, 15, 35, 45, 55, 65, 85, etc.) you also eat at Subway.
3) On all other days, you eat at Lazy Moon.

Write a program that prompts the user to enter a day (which will be a positive integer) and prints out whether your dinner on that day will come from Subway or Lazy Moon.

Input Specification

1. The input day will be a positive integer less than or equal to 366.

Output Specification

Output a statement with one of the two following output formats:

```
On day X, you will eat at Lazy Moon.
On day X, you will eat at Subway.
```

where X is the day number entered by the user.

Sample Run

```
What are day's dinner schedule do you want?
48
On day 48, you will eat at Subway.
```

Problem C: Biology Department

Your friends are getting mad at you for working for Parking Services because they are not happy with all the parking tickets they are receiving. Although it's certainly not your fault, you've decided that life would be easier without their constant teasing. You quit your job with Parking Services and are very surprised that the Biology department is looking for students to write C programs for them.

One common task that the biology students must undertake is cultivating specimens of bacteria. Typically, each day, the bacteria multiply at some given rate. (For example, if there are b bacteria at the beginning of the day, there will be 1.1b bacteria at the end of the day.) But, some of these bacteria, a constant number, also die. Thus, the formula for figuring out how many bacteria are alive on day n+1 is based on how much bacteria are alive on day n in the following manner:

$f(n+1) = bf(n) – d$

where f(n) represents the number of bacteria alive at the beginning of day n, b represents the growth rate (a real number greater than 1), and d represents the number of bacteria that die each day.

The biology department would like you to write a program that takes in the following information:

1) Number of bacteria on day 1
2) Growth rate
3) Number of bacteria that die each day
4) Number of days the bacteria will grow

And outputs a chart to the screen that shows how many bacteria will be alive on each day.

Note: If the number of bacteria alive after replication is less than d, the number that are supposed to die each day, then all remaining bacteria die. (You can't have a negative number of bacteria.)

Furthermore, in all cases, there is one of three outcomes:

1) The bacteria continue to indefinitely grow (well, in real life the lab will just kill them)
2) A steady state is achieved with the same number of bacteria each day
3) The number of bacteria alive will eventually get down to 0.

You should also output which of these scenarios will happen.

Input Specification

1. The number of bacteria of day 1 will be a positive integer.
2. The growth rate will be a real number greater than 1 and less than 10.
3. The number of bacteria that die each day will be a positive integer.
4. The number of days the bacteria will grow will be a positive integer greater than 2 and less than 100.

Output Specification

Output a chart with the following column headers:

```
Day        Number of Bacteria
```

Each following line should have the following format:

```
d          X
```

where d represents the day number of growing the bacteria, and X represents the number of bacteria alive at the beginning of that day. (You'll need to store the number of bacteria in a double, but print it out as an integer by casting the variable. If necessary, ask your instructor how to do this.)

After the chart, output a blank line and then output one of the following statements:

```
The bacteria will eventually die out.
The bacteria will live in a steady state forever.
The bacteria will grow without bound.
```

depending on which of the three scenarios is accurate.

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 bacteria are there on day one?
1000000
What is their growth rate?
1.1
How many bacteria die each day?
150000
How many days will you grow the bacteria?
5

Day       Number of Bacteria
1         1000000
2         950000
3         895000
4         834500
5         767950

The bacteria will eventually die out.
```

Deliverables

You must submit following three .c files:

1) *permit.c*, for your solution to problem A
2) *dinner.c* for your solution to problem B
3) *bacteria.c* for your solution to problem C.

over WebCourses.

Restrictions

Although you may use other compilers, your program must compile and run using DevC++. Please use DevC++ to develop your program. Your program should include a header comment with the following information: your name, course number, section number, assignment title, and date. Also, make sure you include ample comments throughout your code describing the major steps in solving the problem.

Grading Details

Your program 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.)
3) Compatibility to DevC++. (If your program does not compile in either of these environments, you will get a **sizable** deduction from your grade, likely to be over 50%)