CISC 105 Fall 2006 Project 1

Due Sunday Oct 22 at midnight

on MyCourses, paper at the start of class the next day.

This small project is about using loops to perform calculations.

There are formulae available to calculate compound interest and triangular numbers directly; do not use them. This exercise is to help you learn to use loops to perform tasks that require modifying quantities over time. If you have questions about how the task is to be performed, first read and re-read the instructions below; then ask your TA or instructor.

Be very careful about whether things happen at the end of the month or the beginning. This is where most errors crop up for the financial calculations.

Part One: Compound Interest

Terms: interest rate, principal

Principal in this case is the money you put into the bank to earn interest.

Now the interest rate is the amount a bank is paying you for the use of your money. It is stated as an annual rate, but interest will be calculated and added to your principal every month, so that it grows faster; this is called "compounding" (most banks actually compound interest daily).

For this problem, we will examine how money grows using interest compounded monthly.

Use a simple loop. Every month, compute a month's worth of interest and add it to the principal. Show the output as below:

```
> a.out
Enter a principal amount:
1000
Enter an annual interest rate:
7
Your first month of interest will be: 5.83
Enter the number of years your money will be in the bank:
30
You plan to deposit $1000.00 for a term of 30 years.
The total amount of interest you earn will be $7116.50.
Your final account balance will be $8116.50.
>
```

Part Two: Triangular numbers

Consider the following:

```
*               1               1
**              3               4
***             6               9
****            10              16
*****           15              25
```

The nth triangular number is the sum of all integers between 1 and n. Intuitively, it is the number of asterisks in the triangle shown. The third triangular number is 6.

The sum of any two consecutive triangular numbers is a perfect square.

Your function will print (as shown) n asterisks, the nth triangular number, and the nth perfect square. It will take as arguments start and end, so that you can print only some data. Here is the output when the function is called on a start of 5 and end of 8:

```
*****           15              25
******          21              36
*******         28              49
```

The Menu

Once you have two programs working perfectly, you will develop a third program. This one will have two functions using the same code as your first two programs. Then use a switch statement to allow a user to select which function to perform. If the user chooses the interest function, the function will be called and will ask the user for data.

However, if the user asks to compute triangular numbers, the user will be asked for start and end before control is transferred to the function.

Finally, enclose the whole switch statement in a sentinel loop so that the user can run the program as many or few times as they wish.

Submission

Design and test your program carefully using data that you make up. Check the calculations! Friday before the project is due you will be given test data and a project submission sheet. You must show that your third program performs correctly for all test data, and you must complete the **submission sheet** and turn it in with your project's **paper copy** (when?). Of course, code for all **three** programs must be submitted to **MyCourses** along with your script file.