CISC 105 Spring 2006 Project 1

Due Thursday, March 16th at midnight

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

This project is about using loops to perform calculations.

There are formulae available to calculate loan payments and compound interest 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: Loan payments

Terms: interest rate, principal, payment

The interest rate is given to you as an annual rate; you have to convert it to a monthly rate to use it.

The total amount you owe is called the principal balance of the loan.

A payment is money you pay to the bank monthly. Each payment is applied to both interest and principal.

Every month you must pay all the interest owed on the principal balance. Any additional money that you pay will be used to reduce the principal.

For example, suppose you borrow $10,000 at 12 percent. That means that for the first payment (at the end of the first month), you will owe one month of interest on exactly 10,000 dollars, or .01 * 10000 = $100.

If your monthly payment amount is $150, then the first $100 will go to pay interest, and the remaining amount, $50, will be applied to the principal (used to reduce the principal amount).

That means that for the second month, you will now owe a month of interest on $9,950, or $99.50, and so your $150 payment will now contribute $50.50 to paying off principal. Every month the amount of interest changes, and since the payment amount is fixed, the amount applied to principal also changes.

A larger monthly payment would reduce your principal faster.

Use a simple loop to count the number of months it takes to pay off a loan, given a fixed interest rate, principal amount, and payment amount.

Use type double for decimal values. Only print out two decimal places, and for this exercise do not worry about rounding errors (though they would be of concern in a real interest calculation program!).

Your program should produce the following output:

```
> a.out
Enter a principal amount:
10000
Enter an annual interest rate:
12
Your first month of interest will be: 100.00
Enter the amount of your monthly payment:
200
You borrowed $10000.00
Your payment schedule consists of 69 payments of $200.00
and a final principal payment of $131.06.
The total amount of interest you pay will be $3931.06.
>
```

Part Two: Regular Deposits

Many savings and retirement plans offer to deduct money from your monthly paycheck. Let's assume you will add money every month. You can think of the money as a monthly payment, and it becomes very much like a loan calculation.

Start with an initial principal amount and a fixed payment amount. At the end of every month, the payment amount is added to the principal, along with the interest that the principal has earned for the past month, so the principal grows in two ways.

Show the following output:

```
> a.out
Enter an initial deposit amount:
2000
Enter an annual interest rate:
7
Your first month of interest will be: 11.67
Enter the number of years your annuity will be in the bank:
30
Enter the amount of your monthly payment:
200
You started with principal of $2000.00,
Your payment schedule consists of 360 payments of $200.00,
The total amount of your payments will be $72000.00.
The total amount of interest you earn will be $186227.19.
At the end of 30 years your annuity will be worth $260227.19.
>
```

Part Three: Cube Roots

Newton developed a nice method for calculating numeric roots (your text has an explanation of the general method). A cube root y for some x is a number such that

$$y^3 = x$$

If you start with some x given to you by the user, and guess that y is 1 to start, then a better guess for y will be

$$y_{new} = \frac{x/y^2 + 2y}{3}$$

Use a loop to calculate better and better guesses. Each time you calculate a new guess, print out the guess and the difference between the cube of the new guess and the user's input. Stop when the difference is smaller than some tolerance entered by the user (but you can use a tolerance of .001 for debugging purposes.)

You may assume positive integer user input, but this method can be made to work for negative numbers as well (what would be different?).

The Menu

Once you have three programs working perfectly, you will develop a fourth program. This one will have three functions without arguments (see 3.4 of your text) using the same code as your first three programs. Then use a switch statement to allow a user to select which calculation to perform. 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! Wednesday before the project is due you will be given test data and a project submission sheet. You must show that your fourth 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 **four** programs must be submitted to **MyCourses** along with your script file.