

## **COP 3223H Friday Problem #2: Loops Practice**

### **Part A: Population Growth (growth.c)**

Populations of animals tend to double over a constant amount of time. (Well, not really, but pretend that's the case for this problem!) Write a program that prints out a chart of a species' population at various years based on the following input information:

- 1) How long (in years) it takes the population to double.
- 2) The current population of the species (in the year 2012).
- 3) How many doubling periods you want the chart to be.

You are guaranteed that all three values entered will be positive integers and that the final population of the species will not exceed the storage capacity of an integer variable.

### **Sample Program Run #1**

How long (in years) does the population take to double?

**15**

What is the current population?

**1000000**

How many doubling periods do you want the chart to be?

**4**

Year	Population
2012	1000000
2027	2000000
2042	4000000
2057	8000000
2072	16000000

Program Notes: Note that this chart has one more row than the number of doubling periods, because it includes the original population. Print out the chart by spacing out the two values on each row with a tab.

### **Part B: Printing Payments and Calculating Total Expenditure (loan.c)**

Write a program that prints out the payment plan for a house. Ask the user to input the amount owed on the house, the interest rate of the loan as a percent per year, and the monthly payment value. Your program should print a chart with a row for each monthly payment. The columns of the chart should be the month (this should be a number), the amount of the payment, and the amount owed after the payment is made. All the numbers in the second column should be the same except for possibly the last value. Also, your program should detect if the monthly payment is too small to ever pay off the loan. If this is the case, simply output an error message instead of the chart.

### **Output Details**

Your chart should print out three pieces of information for each month: the month number starting at 1, the payment value printed to 2 decimal places in dollars, and the amount still owed, printed to 2 decimal places in dollars. Each value should be separated by either one or two tabs (“\t”). Create headers for each column of the chart.

### **Sample Program Run #1**

What is the value left on the mortgage?

**10000**

What is the annual interest rate of the loan, in percent?

**12**

What is the monthly payment?

**500**

Month	Payment	Amount Owed
1	500.00	9600.00
2	500.00	9196.00
3	500.00	8787.96
...		
22	500.00	211.37
23	213.48	0

Calculation explanation: If the annual interest rate is 12%, the monthly rate is 1%. Thus, the interest accrued in the first month is  $\$10000 \times .01 = \$100$  and the loan value is  $\$10,100$  right before payment. After the payment, the amount owed is  $\$10100 - \$500 = \$9600$ . After the second month, the value owed is  $\$9600 + \$9600 \times .01 = \$9696$ . After making the payment,  $\$9196.00$  is owed. In month 22,  $\$711.37$  is owed. After the  $\$500$  payment,  $\$211.37$  is owed. BUT, after the end of month 23,  $\$213.48$  is owed due to the interest on the  $\$211.37$  for one month, which is about  $\$2.11$ . Thus, the last payment has to be  $\$213.48$  and not  $\$211.37$ .

### **Sample Program Run #2**

What is the value left on the mortgage?

**10000**

What is the annual interest rate of the loan, in percent?

**12**

What is the monthly payment?

**100**

Sorry, this plan won't work.

Calculation Explanation: If you accrue  $\$100$  in interest, your payment must be MORE than  $\$100$  to eventually pay down your loan.

**Note:** The numbers in these samples more closely resemble a bad car loan than a house loan. These numbers were chosen so that the chart printed isn't so long and so the math for the first couple steps is relatively easy to trace by hand.

### **Deliverables**

Two source files: *growth.c*, for your solution to problem A and *loan.c* for your solution to problem B. **All files are to be submitted over WebCourses in the appropriate group turn in.**

### **Restrictions**

Although you may use other compilers, your program must compile in gcc and run in the Code::Blocks environment. Each of your two 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. **Make sure to use good programming style, including use of appropriate constants, good variable names and good use of white space. A significant portion of your grade will be based upon programming style and not correctness. Of course, a significant portion of your grade will also be based upon correctness.**