EECS 10: Homework 4

Prof. Rainer Doemer

October 19, 2007

Due Monday, October 29 at 12:00 noon

1. Part 1: Set of numbers (20 points)

Write a C program that takes a set of integers as input from the user and computes the following values:

- Minimum of the set of numbers (the smallest value)
- Maximum of the set of numbers (the largest value)
- Sum of the set of numbers
- Count of the numbers in the set
- Average of the numbers in the set

Your program should stop getting input from the keyboard when -1 is entered by the user. This -1 is not to be considered as a part of the set of numbers.

In your program, you will need to make use of ‘while’ loops and ‘if’ conditions. Also, keep in mind that the computed average is not necessarily an integer.

When executed, your program should look like this:

```
Please enter the numbers (-1 to quit):
10
12
3
7
25
-1
Minimum : 3
Maximum : 25
Sum     : 57
Average : 11.400000
Count    :  5
```

Your program should also properly handle the case where no numbers are entered (i.e. -1 is the first input to the program). In this case, the program should only output the following message:

```
No numbers are input.  Exiting program …
```

You should submit your program code as file **`numbers.c`**, a text file **`numbers.txt`** briefly explaining how you designed your program, and a typescript **`numbers.script`** which shows that you compile your program and run it using the values 10, 12, 3, 7, 25 as inputs.

2. Part 2: Paying off Credit Card Debt (20 points)

Given a credit limit, current balance, annual percentage rate (APR) and monthly payment, write a C program that computes and prints the interest paid and the remaining balance at the end of each monthly cycle. The program should continue printing the monthly values until the debt is paid off (i.e. the remaining balance becomes zero).

Your program should ask the user for the credit limit, balance on the card, APR (in percent) and the monthly payment as inputs in the beginning.
For example, if credit limit = $5000, balance = $2000, APR = 14.99% (floating point value 14.99), and monthly payment = $200, then your program output should look as follows:

```
Enter the credit limit        : 5000.00
Enter the balance on the card : 2000.00
Enter the APR                 : 14.99
Enter the monthly payment     : 200.00

Month   Balance      Interest     Payment      New Balance
 1      2000.00      24.98        200.00       1824.98
 2      1824.98      22.80        200.00       1647.78
 3      1647.78      20.58        200.00       1468.36
 4      1468.36      18.34        200.00       1286.71
 5      1286.71      16.07        200.00       1102.78
 6      1102.78      13.78        200.00        916.55
 7       916.55      11.45        200.00        728.00
 8       728.00       9.09        200.00        537.10
 9       537.10       6.71        200.00        343.81
10       343.81       4.29        200.00        148.10
11       148.10       1.85        149.95          0.00

It will take $2149.95 over 11 months to pay off this debt.
```

Note: For all dollar amounts and the APR value, print out exactly 2 digits after the decimal point. Also ensure that all the numbers in the output table line up nicely so that the decimal points are all at the same column position. Also, for the last month, the payment need only be as much as the remaining debt. In the example, the last month's payment is $149.95 instead of $200.

The first column 'Month' keeps count of the number of months as the remaining debt diminishes each monthly cycle.

The second column 'Balance' is the balance on the credit card at the beginning of each monthly cycle. For the first month, the balance is the value input by the user at the beginning of the program. For subsequent months, the balance is calculated using the following formula:

Balance = ‘New Balance’ from the previous month.

The third column ‘Interest’ is the interest accrued on the ‘Balance’ at the end of each monthly cycle. It is calculated using the formula: *Interest = balance*(APR/100)/12.*

The fourth column is the ‘Payment’ at the end of each monthly cycle. In our program, this will be a constant value that the user inputs at the beginning. In our example, this is $200.

The fifth column is the ‘New Balance’ at the end of each monthly cycle. It is calculated using the following formula: *New Balance = Balance + Interest - Payment*

You should submit your program code as file **`creditcard.c`**, a text file **`creditcard.txt`** briefly explaining how you designed your program, and a typescript **`creditcard.script`** which shows that you compile your program and run it. Use the following inputs to test your program:
Credit limit : $5000
Balance : $2000
APR : 14.99
Payment : $200

3. Bonus Problem [5 Points]

Extend the credit card program in part 2 to calculate over-the-limit charges if the balance on the card is over the credit limit. Over-the-limit charges will be calculated as follows:

Over-the-limit charges = Over-the-limit fee of $50 plus 2% (default rate) of the amount that is over the limit in the given month.
For example, with a credit limit of $2000.00 and a balance of $2500.00, the over-the-limit charges are $60.00 (in the first month).
Be sure to take the over-the-limit charges into account when calculating the new balance.

Note: You do not have to obtain the over-the-limit fee and the default rate as inputs from the user. Please just use the constants $50 and 2%, respectively, as given above.

Use the following inputs to test your program:
Credit limit : $2000
Balance : $3000
APR : 14.99
Payment : $300

Your output should like the one shown below:

```
Month   Balance     Interest    Payment  Over Limit Fee  New Balance
 1     3000.00      37.48      300.00   70.00             2807.48
 2     2807.48      35.07      300.00   66.15             2608.69
 3     2608.69     32.59      300.00   62.17           2403.46
 4     2403.46     30.02      300.00   58.07           2191.55
 5     2191.55     27.38      300.00   53.83           1972.75
 6     1972.75     24.64      300.00    0.00           1697.40
 7     1697.40     21.20      300.00    0.00           1418.60
 8     1418.60     17.72      300.00    0.00           1136.32
 9     1136.32     14.19      300.00    0.00            850.52
10      850.52     10.62      300.00    0.00            561.14
11      561.14      7.01      300.00    0.00            268.15
12      268.15      3.35      271.50    0.00              0.00

It will take $3571.50 over 12 months to pay off this debt.
```

To submit, use the same files as in Part 2, i.e. **`creditcard.c`**, **`creditcard.txt`**, and **`creditcard.script`**.

4. Submission

Submission for the files will be similar to last week’s assignment. Create a directory called hw4. Put all the files for assignment 4 in that directory and run the **`/ecelib/bin/turnin`** command to submit your homework.