Lab 06: Loops

CSE 4108

Structured Programming I Lab

September 2022

Lab Tasks

1. **Reverse Digits 2 - Use of Loops:**

In **Lab 4 - Expressions (Reverse Digits)**, you were asked to write a program that displays a *two-digit* number with its digits reversed. Generalize the program so that the number can have one, two, three, or more digits.

2. **Even Squares:**

Write a program that prompts the user to enter a number *n*, then prints all even squares between *1* and *n*. For example, if the user enters *100*, the program should print the following:

Sample run:

Enter the value of n: 100
4
16
36
64
100

3. **A lot to print:**

Write a program that finds a bunch of things (***number of inputs, sum, average, maximum, number of odd values***)in a series of numbers entered by the user. The program must prompt the user to enter numbers one by one. When the user enters **0** or **a negative number**, the program will end and display the results.

Sample Run:

Enter a number: 5
Enter a number: 10
Enter a number: 11
Enter a number: 1
Enter a number: 3
Enter a number: 8
Enter a number: 10
Enter a number: 4
Enter a number: 1
Enter a number: 6
Enter a number: 7
Enter a number: 10
Enter a number: 7
Enter a number: 0

The number of inputs is 13
The sum of the input values is 83
The average of the input values is 6.38
The number of odd values input is 7
The maximum value entered was 11

4. **Forever A loan**:

ABC Bank Ltd. provides loans to its customers. The customers return the amount by a fixed monthly payment. On each month however, the remaining balance is incremented by the interest rate assigned for each customer. The total balance at the end of each month can be calculated using the following formula:

currBalance = prevBalance - pay + (prevBalance × rate)

Here,

currBalance = Updated Balance

prevBalance = Previous Balance

pay = Monthly Payment

rate = Monthly Interest Rate

Your task is to write a program that calculates the remaining payable amount on a loan after

each monthly payment.

Your program will take the loan amount, yearly interest rate, and ask the user for monthly payments as inputs and show the remaining balance as outputs. When the user enters **0** or **a negative number**, the program will end.

You need to display each balance with two digits after the decimal point.

Sample Run:

Enter amount of loan: 20000.00

Enter interest rate: 6.0

Enter monthly payment: 386.66

Balance remaining after payment 1: $19713.34

Enter monthly payment: 386.66

Balance remaining after payment 2: $19425.25

Enter monthly payment: 0

End of Program!!

5. **E:**

The value of the mathematical constant e can be expressed as an infinite series:

$e = 1 + \frac{1}{1!} + \frac{1}{2!} + \frac{1}{3!} + \ldots$

Write a program that approximates e by computing the value of:

$1 + \frac{1}{1!} + \frac{1}{2!} + \frac{1}{3!} + \ldots + \frac{1}{n!}$

where n is an integer entered by the user.

6. **Is it really Christmas?:**

Given an integer input **n**, build a christmas tree.

Sample run:

Input	Output
1	1 *
2	* *** * *
3	* *** ***** *** *** ***
4	* *** ***** ******* *** *** *** ***

Home Tasks:

1. **Calendars:**

Alex is working under his uncle Mr. Sakamoto. Mr. Sakamoto has asked him to print calendars. Specifically, he wants one-month calendars. Mr. Sakamoto will specify the number of *days* in a *month* and the *day* of the *week* on which the *month* begins.

Alex, being busy with other tasks, has asked you to help.

Sample run:

Enter number of days in month: 31

Enter starting day of the week (1=Mon, 7=Sun): 3

```
Mo Tu We Th Fr Sa Su
       1  2  3  4  5
 6  7  8  9 10 11 12
13 14 15 16 17 18 19
20 21 22 23 24 25 26
27 28 29 30 31
```

Hint: This program isn't as hard as it looks. The most important part is a for statement that

uses a variable i to count from 1 to n, where n is the number of days in the month, printing

each value of i. Inside the loop, an if statement tests whether i is the last day in a week; if so, it prints a new-line character

2. **Compare Dates II:**

In Lab 5 - Selection Statements (compare_dates.c), you were asked to write a program that determines which of two dates comes earlier on the calendar. Generalize the program so that the user may enter any number of dates. The user will enter **0/0/0** to indicate that no more dates will be entered.

Sample run:

Enter a date (mm/dd/yy): 3/6/18
Enter a date (mm/dd/yy): 5/17/17
Enter a date (mm/dd/yy): 6/3/17
Enter a date (mm/dd/yy): 0/0/0
5/17/17 is the earliest date

3. **Never Too Small:**

Modify the task "**E**" so that the program continues adding terms until the current term becomes less than **ε**, where **ε** is a small (floating-point) number entered by the user.