Computer Programming Lab (CSE104L)
Lab Training 7

Q1. In this training, we will do series of tasks to progressively learn features of C functions. He our focus is to learn functional abstraction, i.e., creating functions so that we can use them whenever we want, without repeating the code.

Task 1: Consider the problem of validating a date given by three int values dd, mm, and yyyy. Here is a clean prototype of a function that takes three arguments: dd, mm, yyyy and returns an `int` (Boolean) value to indicate if it is a valid date.

```
int isValid(int dd, int mm, int yyyy);
```

Next, create a definition of the function by adding the details of actions necessary. Please write the declaration (prototype) near the beginning of your .c file.

Task 2: To test the above function, write a main function that runs a loop (for-loop) 10 times. Each time it asks the user to give a date (dd-mm-yyyy) and then calls function isValid() to test and print the validity of each date using printf() function.

Task 3: Write a function prototype and its definition that takes two int arguments: mm and yyyy. It returns the number of days in month mm-yyyy after accounting for issues such as leap year.
Test your function using many pairs of mm and yyyy values.

Task 4: Write function dayOfWeek() based on Zeller's congruence. Its takes date and tells the corresponding day of the week. (see details at the end of the document). Write proper prototype and definition for this function.

Task 5: Write function main() of a program. This program will loop till the user gives the date 00-00-0000 as input. For each date it reads, the program will print the name of the day of the week or print invalid date.

Q2. Write a function `swapNumbers()` that takes two integer numbers, print the initial values of these numbers, swap their values and print the numbers after swapping. Create a main function and call this function `swapNumbers()` for the numbers entered by the user.
Your main() function and `swapNumbers()` should not contain any `printf()` and `scanf()` statements. You need to design separate functions for reading the values from the keyboard and printing the output.

Q3. Write a function that will round a floating point number to an indicated decimal place. For example the number 17.457 would yield the value 17.46 when it is rounded off to two decimal place.

Q4. Write a C program to convert decimal number to binary number using the function. The function reads the decimal number and print it's binary equivalent.

Day of the week for a date

Wikipedia.org describes a way to find weekday for any date using Zeller's congruence. A useful version of the congruence suitable for Gregorian calendar for computer implementation is expressed by formula:

$$h = \left(q + \left\lfloor \frac{13(m+1)}{5} \right\rfloor + Y + \left\lfloor \frac{Y}{4} \right\rfloor - \left\lfloor \frac{Y}{100} \right\rfloor + \left\lfloor \frac{Y}{400} \right\rfloor\right) \bmod 7,$$

where

- h is the day of the week (0 = Saturday, 1 = Sunday, 2 = Monday, ..., 6 = Friday)
- q is the day of the month
- m is the month (3 = March, 4 = April, 5 = May, ..., 14 = February)
- Y is modified year, which is (year -1) during January and February.

```
#include <stdio.h>
#include <stdlib.h>

int main(int argc, char *argv[]) {
   int dd = 19, mm = 7, yyyy = 2019;
   int h, m, mod_yyyy;
   if (mm <3)
   {
         m = mm + 12;
         mod_yyyy = yyyy - 1;
   }
   else
   {
         m = mm;
         mod_yyyy = yyyy;
   }
   h = (dd + 13*(m+1)/5 + mod_yyyy + mod_yyyy/4 -
   mod_yyyy/100 + mod_yyyy/400) % 7;
   switch (h)
   {
         case 0:
               printf("Date %d-%d-%d is a Saturday\n", dd, mm, yyyy);
               break;
         case 1:
               printf("Date %d-%d-%d is a Sanday\n", dd, mm, yyyy);
               break;
         case 2:
               printf("Date %d-%d-%d is a Monday\n", dd, mm, yyyy);
               break;
         case 3:
               printf("Date %d-%d-%d is a Tuesday\n", dd, mm, yyyy);
               break;
         case 4:
               printf("Date %d-%d-%d is a Wednesday\n", dd, mm, yyyy);
               break;
         case 5:
               printf("Date %d-%d-%d is a Thursday\n", dd, mm, yyyy);
               break;
         case 6:
               printf("Date %d-%d-%d is a Friday\n", dd, mm, yyyy);
   }
}
```