Programming for Engineers

Fall 2021-2022

Assignment 1

STUDENT INFORMATION					GRADE
Session No	Student No	Name and Surname	Dept.	Signature	
1	64180006	Gülsüm İkbal Avşar	CoE - EEE		

Problem-1

The question wants from us to write a program that when we enter the year, it will print the 12 months of this year. Besides, leap years should have been considered.

Write a C/C++ program that will take the year as input and will print the 12 months calendar as output. Do not forget to take leap years into consideration. Concerning the format of the output, an example is given next:

October 2021

Sun	Mon	Tue	Wed	Thu	Fri	Sat
					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						

- **The Program Code**

```
#include <iostream>
#include <iomanip>

using namespace std;

int day_numbers(int day, int month, int year);
string MonthName(int monthNumber);
int DaysOfMonths(int month, int year);
void calendar(int year);

int main() {
    int year; // we will get the year from user.

    cout<<"Enter year that want to display: ";
    cin>>year; // reading year
    calendar(year); // displaying 12 months calendar

    return (0);
}
```

```
int day_numbers(int day, int month, int year){
    static int t[] = {0,3,2,5,0,3,5,1,4,6,2,4};
    year -= month < 3 ;
    return ( year + year/4 - year/100 + year/400 + t[month -1]) % 7;
}

// To return the month name, we need this function. We get the month number as
// parameter, and using the array of months we can return the expected month form this array.
string MonthName(int monthNumber){
    string month;
    string allMonths[] = {"January", "February", "March","April",
                          "May","June","July","August","September","October"
                          ,"November","December"};
    month = allMonths[monthNumber];
    return (month);
}
```

```
// We need to find that whether a month is 30 days or 31 days.
// Besides, we should check it is a leap year or not because of February.
// That's why we get the year as an input with month number.
int DaysOfMonths(int month,int year){
    if(month == 0 || month == 2 || month == 4 || month == 6 || month == 7 || month == 9 || month == 11){
        return (31);
    }else if(month == 1){ // When we display February, we should check whether year is a leap year or not.
        if(year % 4 == 0){
            if(year % 100 == 0){
                if(year % 400 == 0){
                    return(29);
                }else{
                    return(28);
                }
            }else{
                return(29);
            }
        }else{
            return(28);
        }
    }else{
        return(30);
    }
}
```

```
void calendar(int year){
    cout << "---- Calendar of "<< year <<" ----"<< endl;
    int dayValue; // to keep the day number of a month.
    int current = day_numbers( day: 1, month: 1,year) + 1;

    for(int i = 0; i <= 11; i++){   // This for loop provide us to determine the month and display the days of it
        dayValue = DaysOfMonths(i,year);
        cout << endl <<"           "<< MonthName(i)<<" "<<endl;
        cout <<" Sun  Mon  Tue  Wed  Thu  Fri  Sat"<< endl;
        cout<<" -----------------------------------"<< endl;

        int k=0;
        while(k< current){ // This while loop for arranging the indentation of calendar
            cout <<"     "; // For example, if a month is beginning in Wednesday, days before
            k++;            // Wednesday should be empty.
        }

        for(int j = 1; j <= dayValue; j++)
        { // With this for loop, we can display the days of current month.
            cout << setw( n: 5) << j; // setw function is used for setting the field width. With this code line, we can display
            int a = ++k;            // the days and days number one under the other properly.
            if(a >6)
            { //To pass the new line in a month.
                k = 0;
                cout << endl;
            }
        }
        if(k)
        {
            cout << endl;
        }
        current = k;
    }
}
```

- **A part of Output**

```
          September
 Sun  Mon  Tue  Wed  Thu  Fri  Sat
 -----------------------------------
                   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

          October
 Sun  Mon  Tue  Wed  Thu  Fri  Sat
 -----------------------------------
                             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

          November
 Sun  Mon  Tue  Wed  Thu  Fri  Sat
 -----------------------------------
        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
```

Problem-2

Write a C/C++ program that finds and prints all **prime numbers**, *pn*, that satisfy the following equation:

$$pn = 3^k - 2^k$$

for non-negative integers *k*, (*k*<32).

- **The Program Code**

```cpp
#include <iostream>

using namespace std;

int powerResult(int power); // to calculate the power values.

int main() {

    for(int k = 0; k < 32 ; k++){
        int pn2 = 1;
        int check = 0; // to specify the result is prime or not
        pn2 = powerResult(k);
        // After we calculated the result of subtraction, we should check whether the result is prime or not.
        for(int i = 2; i < pn2 ; i++){
            if(pn2 % i == 0){
                check = 0; // if the result is able to divide for a value between 2 and itself, it is not prime. So, loop will finish.
                break;
            }else{
                check = 1; // if result is not prime, check will be 1.
            }
        }
        if(check == 1){ // to determine the prime values, we look at the check value and display it.
            cout << pn2 << endl;
        }
    }
    return 0;
}
```

```cpp
// Calculating 3^k and 2^k will be performed by this function.
// The parameter power is k which came from main module above.
// Also, after we calculated the 3^k and 2^k, the next operation which is subtraction
// is calculated in this function. We returned the integer result to main module.
int powerResult(int power){
    int pn = 0;
    int value3 = 1 ;
    int value2 = 1;

    for(int i = 0; i < power ; i++){
        value3 *= 3;
        value2 *= 2;
    }
    pn = value3 - value2;
    return pn;
}
```

- **Output**

```
C:\Users\ikbal\CLionProjects\As1Q2_test\cmake-build-debug\As1Q2_test.exe
5
19
211
129009091
1868321459
1312094233
1146497699

Process finished with exit code 0
```