SRM
INSTITUTE OF SCIENCE & TECHNOLOGY
(Deemed to be **University** *u/s 3 of UGC Act, 1956)*

SRM INSTITUTE OF SCIENCE AND TECHNOLOGY FACULTY OF ENGINEERING AND TECHNOLOGY

18CSS101J- PROGRAMMING FOR PROBLEM SOLVING

MINI PROJECT REPORT

CALENDAR

Submitted by:

Name	AYUSH KUMAR
Reg. No	RA2111003011875
Branch:	CSE-CORE
Section:	Z2

ABSTRACT

The Calendar Application Presented Here Is a Very Simple Console Application Developed Using C Programming Language. It Is Built Without Using Graphics properties; instead it Utilizes Many Windows Properties To Give The Application A Colourful Look And Feel. It Is Compiled in Code: Blocks Using GCC/Dev C++/VS Code Compiler. In This Project You Can Find Out the Day Corresponding to Given a Date and View the Days and The Dates Corresponding to a Particular Month Year. The Viewers of The Calendar Program Have to Give Only Year, Month And day As Input to Display Calendar Date and Name of Day of Week with Respect to The Normal Calendar.

CONTENTS

CHAPTER 1.

INTRODUCTION

It Displays a nicely formatted calendar of every month. You can find the day by entering the day, month and year. It provides a very simple interface and displays days, dates, months, and years based on the input given by the user. At first, the user need to provide month and year as input. The application displays day corresponding to a given date. It also displays days and dates corresponding to a particular month and year. Months can be navigated using arrow keys. Also ‘n’ and ‘p’ keys can be used to go next and previous month respectively.

Now there is an issue regarding the days in the particular month. January having 31 days, Feb has 28 days and so on. So for this we are declaring the integer array which consists of the days in the particular month. Another issue is leap year if the entered year is a leap year then the days in February are 29. So use conditional statement which finds that the year is a leap or not and if the condition is true then days in February are 29.

This project will give you a basic overview of how to make a calendar application using C Programming.

CHAPTER 2.

OBJECTIVES

Aim of the project of the calendar in C Programming is to develop this project which helps in finding out the day, date and month one wants to know about.

CHAPTER 3.

SYSTEM REQUIRMENT SPECIFICATIONS

A System Requirements Specification that describes the features and behaviour of a system or software application.

3.1. SOFTWARE REQUIREMENTS:

Programming Language used: C

Operating System: Windows 7 (Minimum)

IDE: Any IDE with C runtime environment.

3.2. HARDWARE REQUIREMENTS:

Hard Disk: 1TB (Minimum)

Processor: Intel Core i3 (Minimum)

CHAPTER 4.

SYSTEM DESIGN

The design of the project is based on C language operations.

We will start creating the input screen first in **main ()** function, after that we used some of the function as mentioned: **textcolor ()** function – It is used to change the color of drawing text in c programs. **printf** function - It is used to show output on the screen. **scanf** function - It is used to take input from the user.

Integer type - The most natural size of integer for the machine.

Character type - Typically a single octet (one byte). It is an integer type. **If Statement for loop -** It is used to iterate the statements or a part of the program several times. It is frequently used to traverse the data structures like the array and linked list.

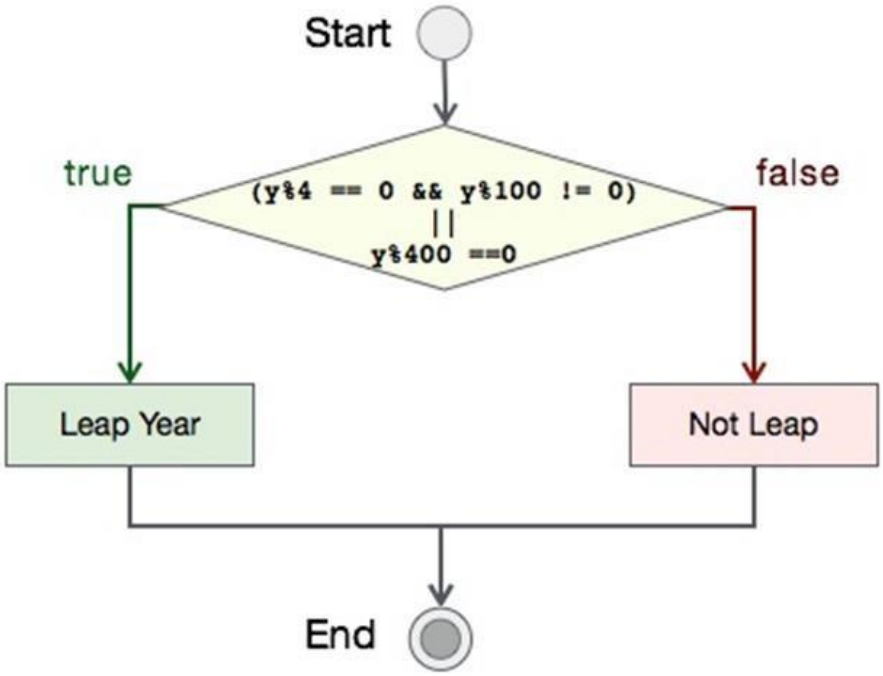


Fig. 4.1. Flowchart of a year leap year or not

Algorithm
Step1: Start
Step2: Read the value of year
Step3: IF year % 4 ==0 THEN
print It is a Leap year
ELSE
print It is not a Leap year
Step4: Stop

Fig. 4.2. Algorithm

CHAPTER 5.

SYSTEM IMPLEMENTATION

The project is implemented as follows:

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

int get_1st_weekday(int year){

  int d;
  d = (((year - 1) * 365) + ((year - 1) / 4) - ((year - 1) / 100) + ((year) / 400) + 1) % 7;
  return d;
}
```

Fig. 5.1.

```c
int main()
{
   system("Color 3F");
   int year,month,day,daysInMonth,weekDay=0,startingDay;
   printf("\nEnter your desired year:");
   scanf("%d",&year);

   char *months[]={"January","February","March","April","May","June","July","August","September","October","November","December"};
   int monthDay[]={31,28,31,30,31,30,31,31,30,31,30,31};

   if((year%4==0&&year%100!=0)||year%400==0)
       monthDay[1]=29;

   startingDay=get_1st_weekday(year);

   for(month=0;month<12;month++){

      daysInMonth=monthDay[month];
      printf("\n\n--------------%s------------------\n",months[month]);
      printf("\n  Sun  Mon  Tue  Wed  Thurs  Fri  Sat\n");

      for(weekDay=0;weekDay<startingDay;weekDay++)
        printf("     ");

      for(day=1;day<=daysInMonth;day++){
        printf("%5d",day);
```

Fig. 5.2.

```
37
38              if(++weekDay>6){
39                  printf("\n");
40                  weekDay=0;
41              }
42              startingDay=weekDay;
43          }
44
45      }
46
47
48  }
```

Fig. 5.3.

CHAPTER 6.

RESULTS

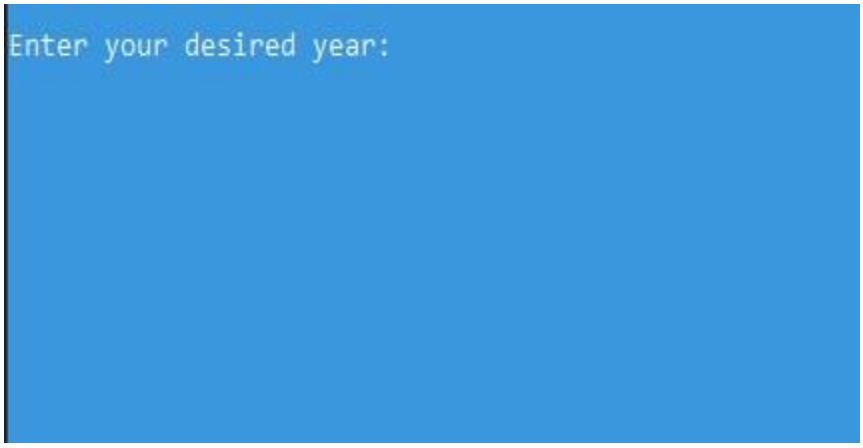
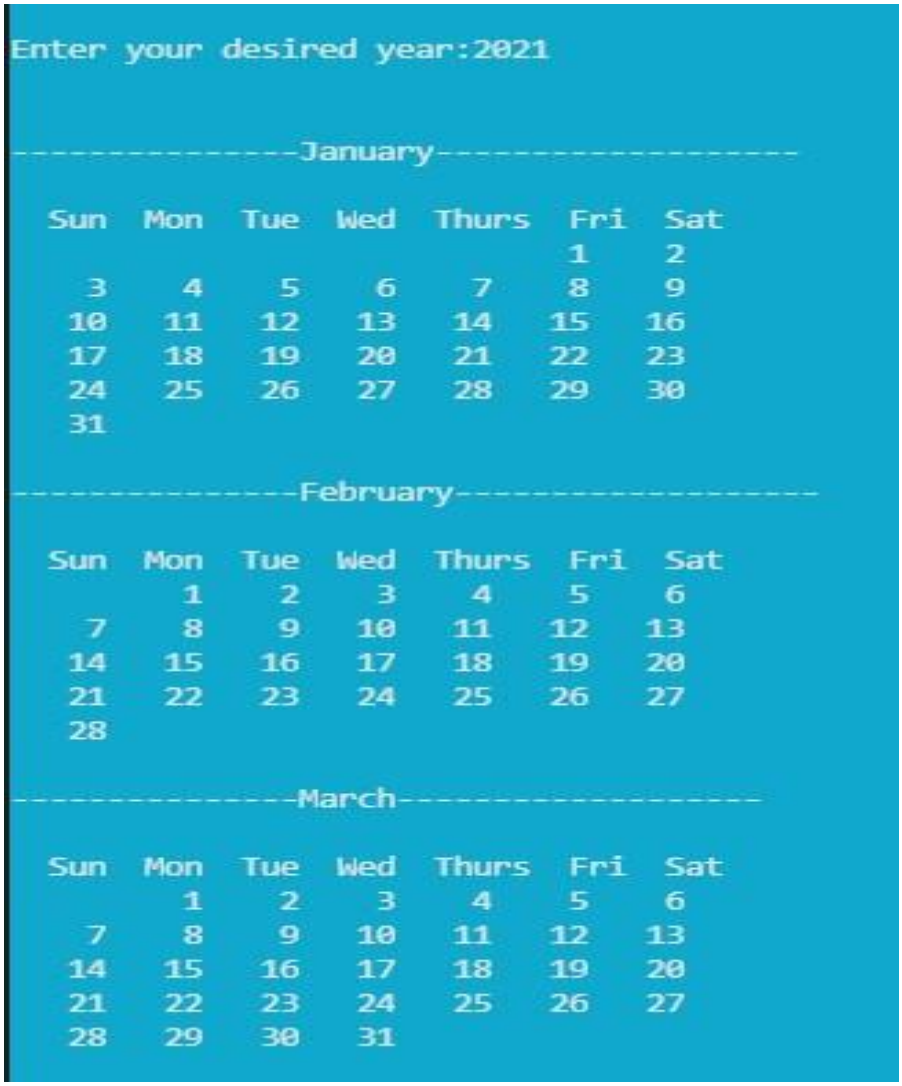


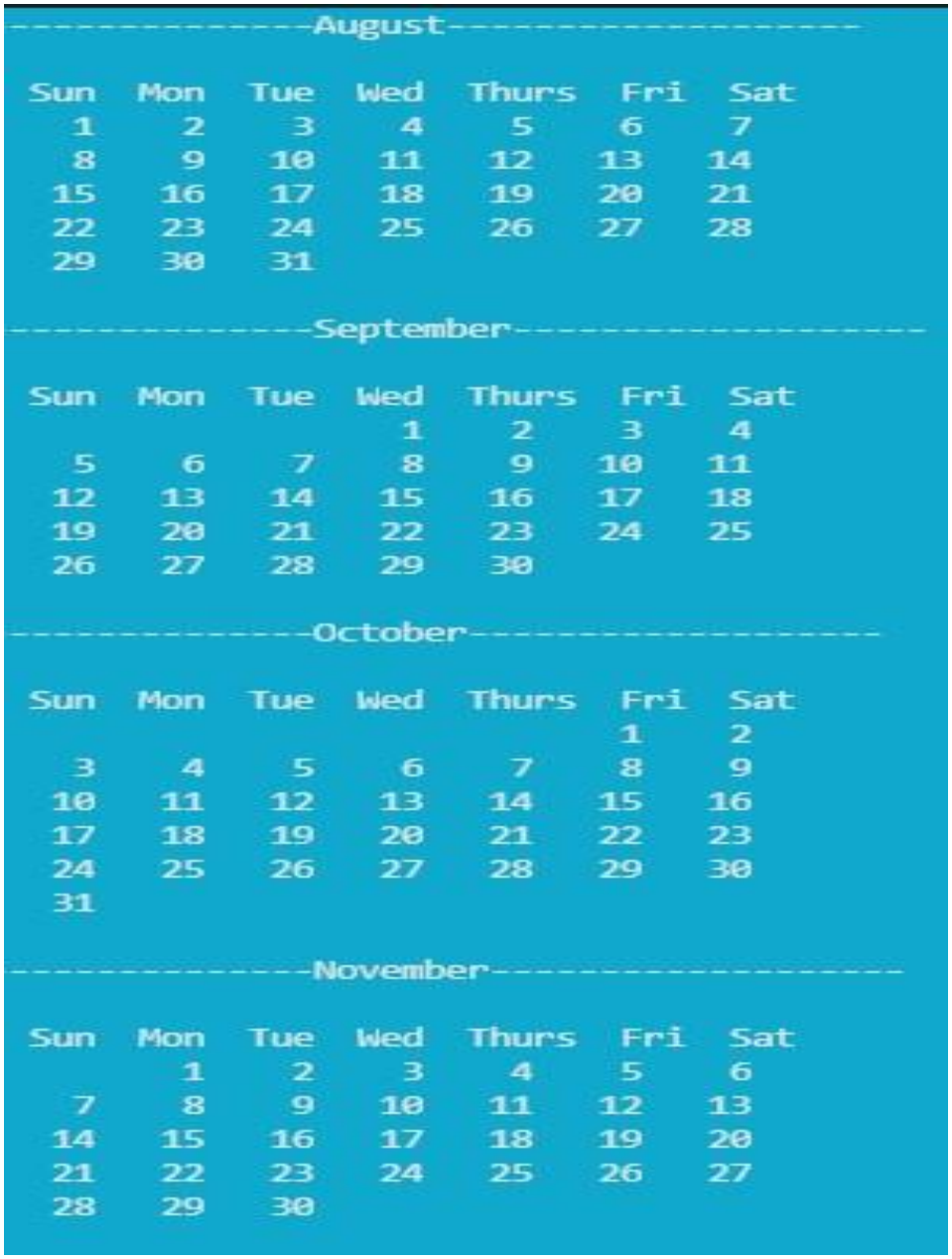
Fig. 6.1 Main menu options

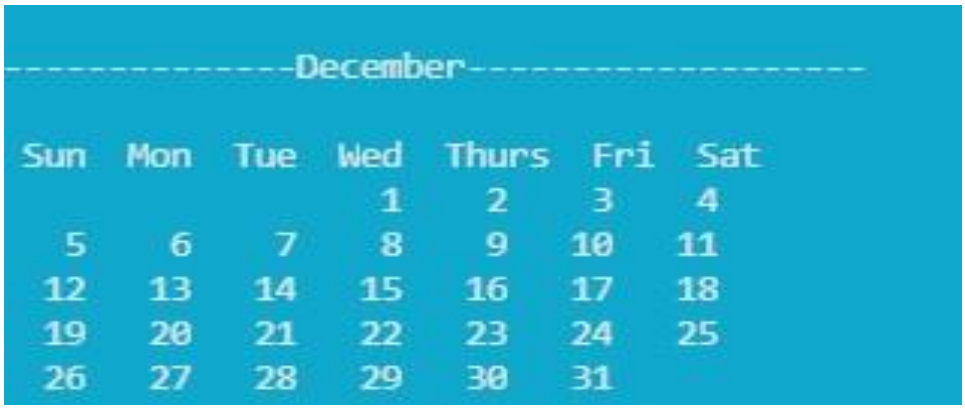
Fig. 6.2 User input

Fig. 6.3. Desired Output




----------April----------
Sun Mon Tue Wed Thurs 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
----------May----------
Sun Mon Tue Wed Thurs 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
----------June----------
Sun Mon Tue Wed Thurs 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
----------July----------
Sun Mon Tue Wed Thurs 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


August
Sun Mon Tue Wed Thurs 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
September
Sun Mon Tue Wed Thurs 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 Thurs 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 Thurs 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


December
Sun Mon Tue Wed Thurs 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 END