**DATA STRUCTURES PROJECT**

**TOPIC: CALENDAR**

**TEAM MEMBERS:**

1. Mehak Katra - 26

2. Bhavesh Khubnani - 27

3. Jairaj Khushalani - 28

4. Devin Lilaramani - 30

1. **PROBLEM STATEMENT:**

To write a C program to implement a calendar which Display Days of a month when year is given.

1. **PROBLEM DECOMPOSITION:**

First we make two arrays; one with the number of days for a given month and one with all the month names. Note: in both arrays the first position is empty on purpose, we want to use 1 to 12 to keep things simple.

**int daycode(int year):**

We have first defined a function to determine the day\_code of the given year.

To solve the given problem,i.e to print the calendar of a given year/month, we use a formula known as Daycode.

Daycode finds the first day(code) of the given year and then using that as basis whole calendar can be printed.

Daycode formula :

da1 = (year-1.0) / 4.0

da2 = (year-1.0) / 100.0

da3 = (year-1.0) / 400.0

Daycode = (year + d1 - d2 + d3) % 7

Daycode is 0 for Sunday, 1 for monday and so on…

For eg:

Year : 1998

da1= 499.25

da2= 19.97

da3= 4.99

Daycode = 2482.27 % 7 = 4.

Therefore Daycode = 4; Hence The first day of the year 1998 will be on Thursday.

After daycode is acquired,using simple loops we can print Dates of a month and hence Print whole calendar for a given year.

**int leap(int year):**

Another important condition to be checked is whether the given year is a leap year or not. Hence, we defined a function to determine whether the year is a leap year.

## **Gregorian Calendar and Leap Years**

The Gregorian calendar is the internationally accepted calendar. In the Gregorian calendar there are leap years. If you take a period of four hundred years, there are 303 normal years and 97 leap years. Most people think that every fourth year is a leap year, but strictly speaking this isn’t true.

## **How to determine which leap years**

If a year is divisible by 4, then it is a leap year. But if that year is divisible by 100 only, then it is not a leap year. However, if the year is also divisible by 400(and 100), then it is a leap year. So we can construct the following statement:

if(year% 4==FALSE&&year%100!=FALSE||year%400==FALSE)

{

monthDays[2]=29; //It is a leap year and February has 29 days.

}

Else

{

monthDays[2]=28; //It is not a leap year and February has 28 days.

}

**void calendar(int year, int day\_code):**

We pass the given year and the day code to this function. Void calender is used to display the required output .The first for loop is used to loop through all months. We then print the month’s name and all the days of the week. We then use the daycode to position the prompt under the right weekday. The next for loop is used to print all the dates for one month. The last thing we do is to set the position of the prompt on the right weekday.

**int main(void):**

The main function is basically used to call all the other functions and also to take the input of the Year.

1. **DATA STRUCTURES USED:**

**monthDays[]-**

Arrays are a type of data structure that can store a fixed-size sequential collection of elements of the same data type.

Instead of declaring individual variables an array is used.

An array is used to store a collection of data and since we had to represent days in a month which is already known to us an array is the best data structure available.

**char \*m[]-**

All the respective months are clubbed together in this array which improves usability and saves time since we dont have to declare multiple variables.

1. **ALGORITHMS:**

**Int daycode(int year):**

**S1:Start**

**S2:**calculate da1=(year-1)/4, da2=(year-1)/100,da3=(year-1)/400

**S3:**calculate day\_code=(year+da1-da2+da3)%7

**S4:**return day\_code

**S5:Stop**

**Int leap(int year)**

**S1:Start**

**S2:**if year%4 is false and year%100 is false or year%400 is false then goto s3 or else got s4

**S3:**set monthDays[2]=29 //monthDays[] is a global array

Goto S5

**S4:**set monthDays[2]=28

**S5:Stop**

**Void Calendar(int year,int day\_code)**

**S1:**Start

**S2:**set month=1 and d=1 repeat S3 to s12 while month<=12

**S3:**print m[month] //m[] is char array for months,goto next line

**S4:**Print names of the days starting from sunday

Print new line

**S5:**set d=1 repeat S6 till d<=1+day\_code\*5

**S6:**print “”

Set d=d+1

**S7:**set d=1 repeat S8 to S11 til d<=monthDays[month]

**S8:**print d

**S9:**if( d+da\_code%)7>0

Print “”

And goto S11

**S10:**Goto a new line

**S11:**set d=d+1

**S12:**set day\_code=day\_code+monthDays[month]

Print new line

**S13**:Stop

1. **IMPLEMENTATION:**

**=> C CODE:**

#include<stdio.h>

#define TRUE 1

#define FALSE 0

int monthDays[]={0,31,28,31,30,31,30,31,31,30,31,30,31};

char \*m[]=

{

" ",

"JANUARY",

"FEBUARY",

"MARCH",

"APRIL",

"MAY",

"JUNE",

"JULY",

"AUGUST",

"SEPTEMBER",

"OCTOBER",

"NOVEMBER",

"DECEMBER"

};

int daycode(int year)

{

int day\_code;

int da1, da2, da3;

da1 = (year - 1.)/ 4.0;

da2 = (year - 1.)/ 100.;

da3 = (year - 1.)/ 400.;

day\_code = (year + da1 - da2 + da3) %7;

return day\_code;

}

void calendar(int year, int day\_code)

{

int month, d;

for ( month=1;month<=12;month++ )

{printf("\n\n");

printf("%s", m[month]);

printf("\n\nSun Mon Tue Wed Thu Fri Sat\n" );

for (d=1;d<=1+day\_code\*5;d++)

{

printf(" ");

}

for (d=1;d<=monthDays[month]; d++ )

{

printf("%2d", d );

if ( ( d + day\_code ) % 7 > 0 )

printf(" " );

else

printf("\n" );

}

day\_code = ( day\_code + monthDays[month] ) % 7;

printf("\n\n");

}

}

int leap(int year)

{

if(year% 4==FALSE&&year%100!=FALSE||year%400==FALSE)

{

monthDays[2]=29;

return TRUE;

}

else

{

monthDays[2]=28;

return FALSE;

}

}

int main(void)

{

int year, day\_code, leap\_year;

printf("Please enter a year : ");

scanf("%d", &year);

printf("");

day\_code =daycode(year);

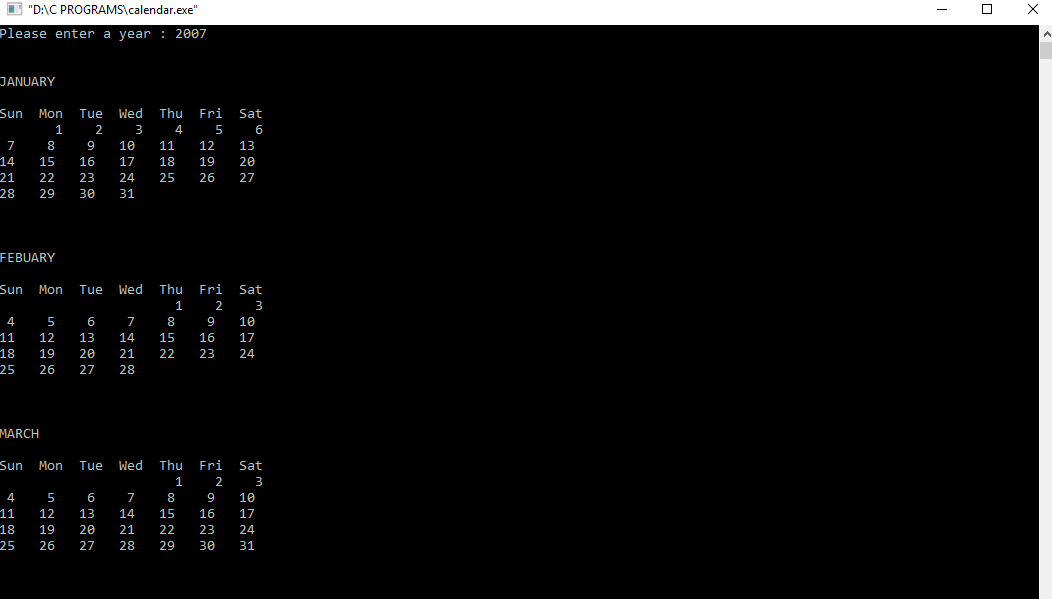
leap(year);

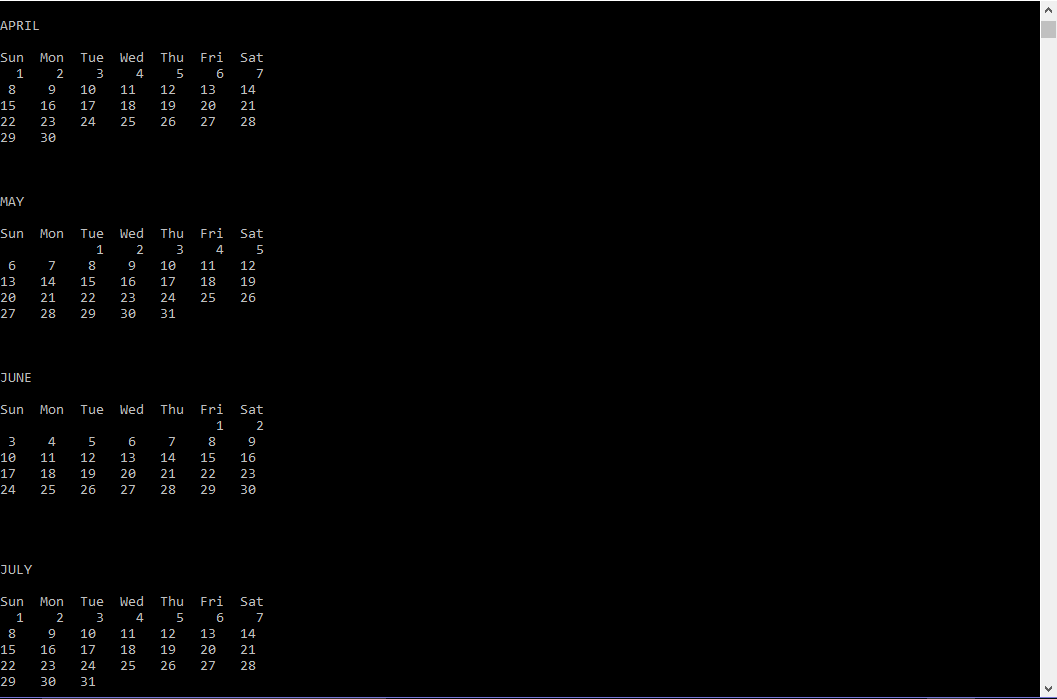
calendar(year, day\_code);

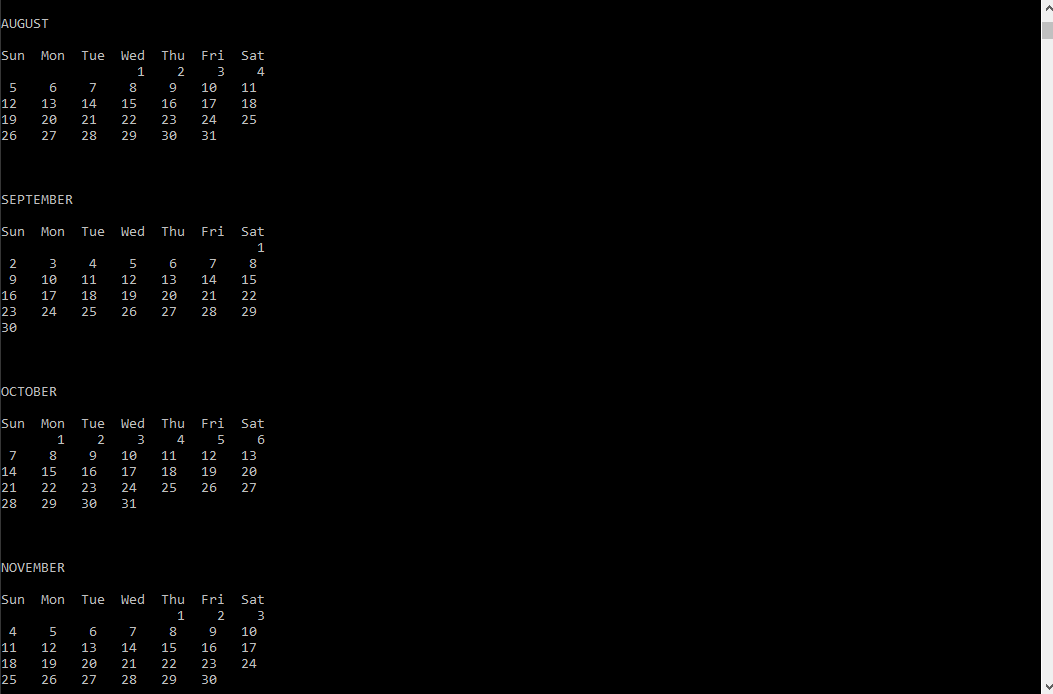
printf("\n");

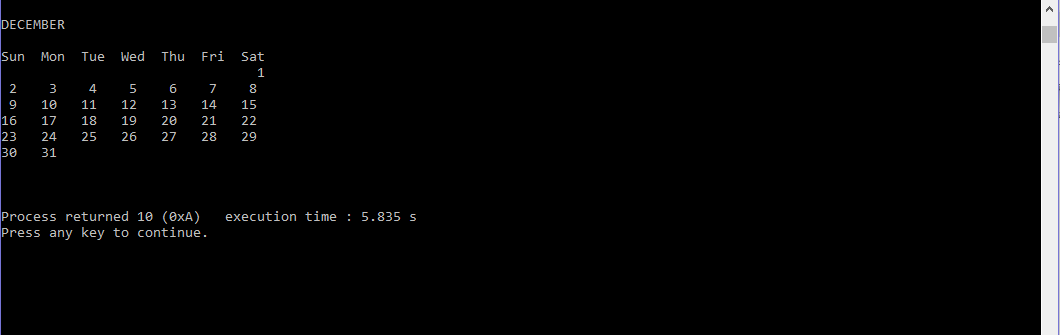
}

**OUTPUT:**

****

****

****

****

**F. CONCLUSION:**

This report has discussed the development of a calendar. The calendar application presented here is a simple application developed using c programming language, which has been built without using any graphical properties. While implementing this project we have learned about the application of arrays,concept of functions and method approach of problem solving (debugging functions, logical errors,etc). It is compiled in Code::Blocks IDE with GCC compiler. Each data structure has its own advantage and disadvantage and must therefore be used according to the application.

By completing this project, we have a deeper understanding towards the importance of arrays which are among the simplest and most important data type used in almost every C program.