# ***REPORT OF***
# ***MINI PROJECT ON***
# ***CALENDAR APPLICATION***

*Submitted by :*

*Srinivas B (RA2111026010079)*

*Course : CSE with Specialization in AIML*

# *CALENDAR APPLICATION*

## *OBJECTIVE:*

To create a calendar application using c programming language.

## *WORKING OF OUR SYSTEM:*

- 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.
- The first function inputyear() is used to get the user input. We ask the user to input a year. Note: that there is no input checking or error handling to keep things simple.)
- The next function determinedaycode() is used to get the day number of the first day in that year, so we can print the date in the correct position. (So it is only used for output purposes.)
- The next function determineleapyear() is used to determine if input of the user is a leap year. If so, the number of days in February is changed to 29. The last function calendar() is used to print each month onto the screen.
- 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. Then we print all the dates for one month. The last thing we do is to set the position of the prompt on the right weekday.

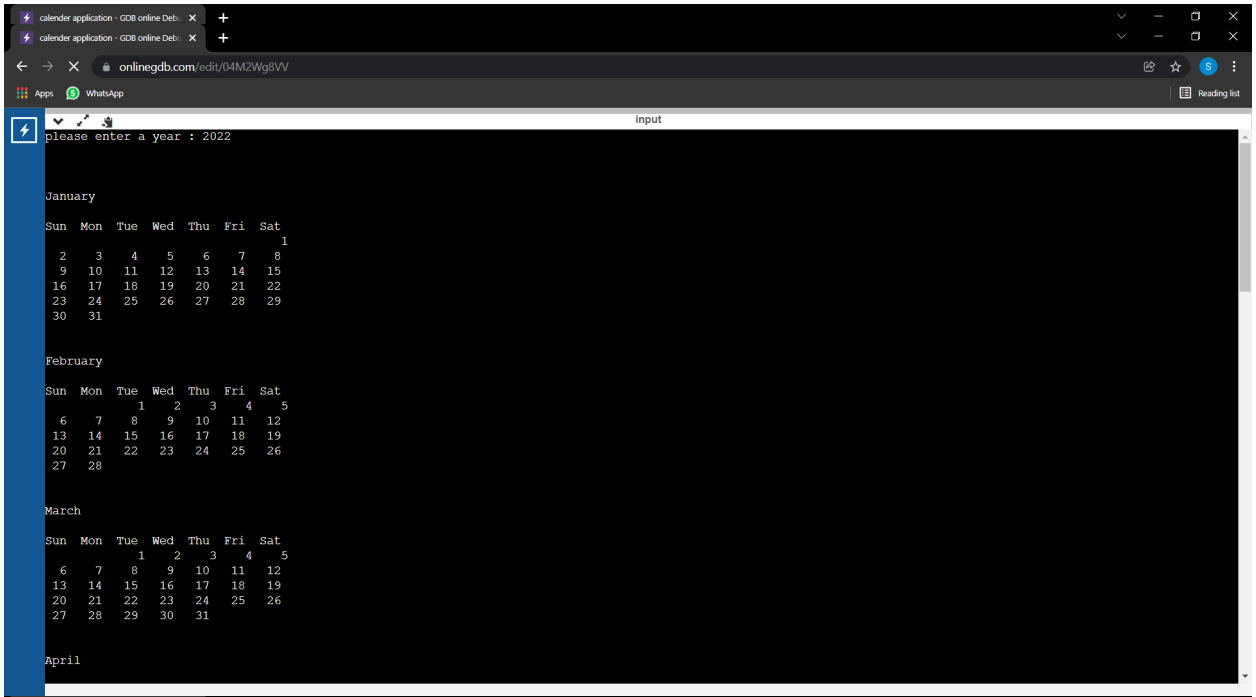
## *INPUT:*

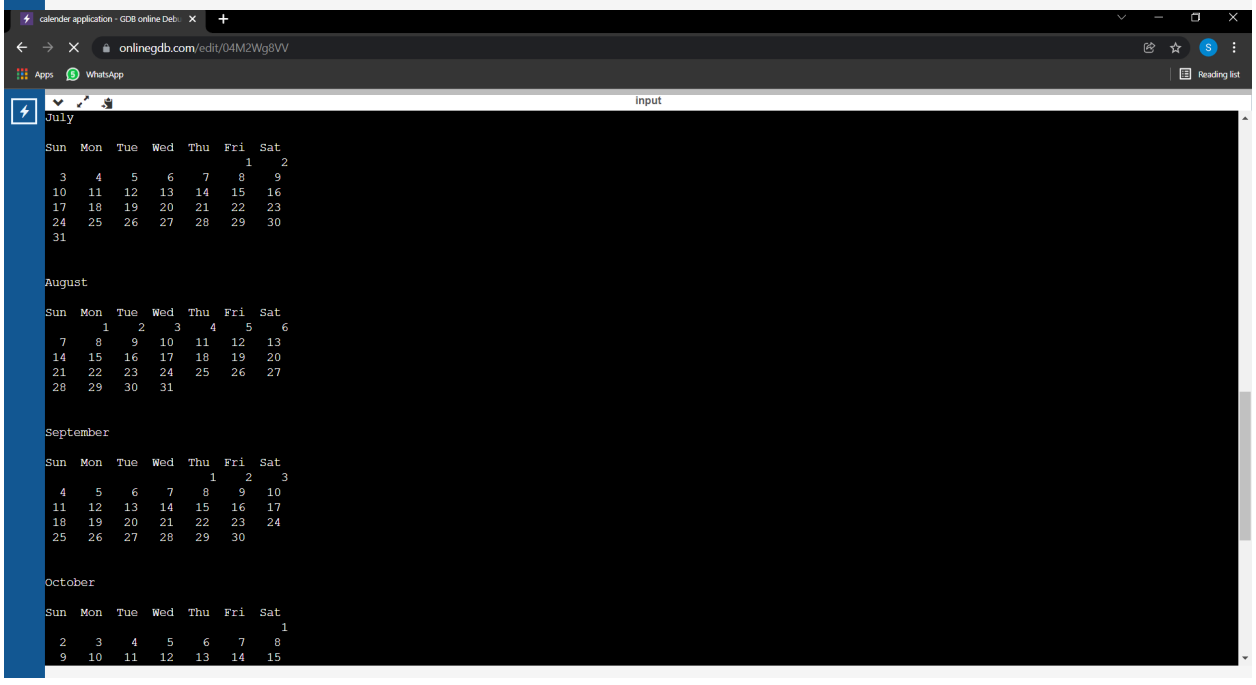
```
#include <stdio.h>
#define TRUE 1
#define FALSE 0

int days_in_month[]={0,31,28,31,30,31,30,31,31,30,31,30,31};
char *months[]=
{
   " ",
   "\n\n\nJanuary",
   "\n\n\nFebruary",
   "\n\n\nMarch",
   "\n\n\nApril",
```

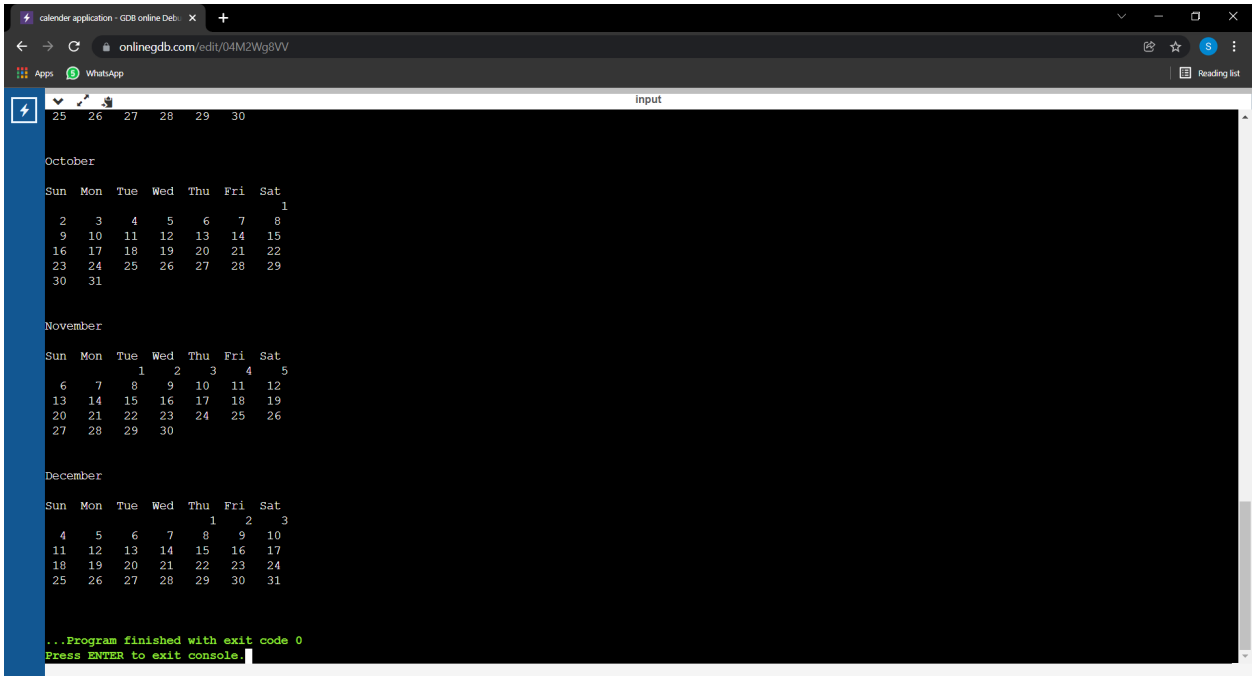
```
    "\n\n\nMay",
    "\n\n\nJune",
    "\n\n\nJuly",
    "\n\n\nAugust",
    "\n\n\nSeptember",
    "\n\n\nOctober",
    "\n\n\nNovember",
    "\n\n\nDecember",
};
int inputyear(void)
{
    int year ;
    printf("please enter a year : ");
    scanf("%d",&year);
    return year;
}
int determineleapyear(int year)
{
    if(year% 4 == FALSE && year %100 != FALSE || year%400 == FALSE)
    {
        days_in_month[2] = 29 ;
        return TRUE;
    }
    else
    {
        days_in_month[2]= 28;
        return FALSE;
    }
}

int deteminedaycode(int year)
{
    int daycode;
    int d1,d2,d3;

    d1=(year -1.)/4.0;
    d2=(year -1.)/100.;
    d3=(year -1.)/400.;
    daycode=(year +d1 - d2 + d3) %7 ;
    return daycode ;
```

```
}
void calendar (int year ,int daycode)
{
   int month ,day ;
   for (month = 1 ; month <= 12; month++)
   {
      printf("%s",months[month]);
      printf("\n\nSun  Mon  Tue  Wed  Thu  Fri  Sat\n ");

      for (day =1 ;day <= 1 + daycode * 5 ;day++ )
      {
         printf(" ");
      }
      for( day =1;day<= days_in_month[month]; day++ )
      {
         printf("%2d ",day);
         if((day + daycode) % 7 >0)
         printf("  ");
         else
         printf("\n ");
      }
      daycode = (daycode + days_in_month[month] ) %7 ;        }
      }
   int main (void)
   {
      int year ,daycode,leapyear;
      year =inputyear();
      determineleapyear(year);
      daycode=deteminedaycode(year);
      calendar(year,daycode);
      printf("\n");
   }
```

*OUTPUT:*


calender application - GDB online Debu
calender application - GDB online Debu
onlinegdb.com/edit/04M2Wg8VV
Apps
WhatsApp
Reading list
input
please enter a year : 2022
January
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
February
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
March
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
April


calender application - GDB online Debu
onlinegdb.com/edit/04M2Wg8VV
Apps
WhatsApp
Reading list
input
July
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
August
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
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




## *CONCLUSION:*

Thus the code is executed successfully and the output is verified. This will be a lot helpful to view all the dates and days of an entire year.