CALENDAR

<u>A Mini Project in C</u>

Academic Year: 2021-22 ODD SEMESTER

Department	: B. Tech Computer science and Engineering with specialization in Artificial Intelligence and Machine Learning.
Semester	: 1
Course Code	: 18CSS101J
Course Title	: Programming for Problem Solving

Submitted by
Shravani Maskar (RA2111026010250)
Aryama Agrawal (RA2111026010242)

Under the Guidance of
Dr. R. Lakshminarayana
(Associate Professor, NWC)

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

DEPARTMENT OF COMPUTING
COLLEGE OF ENGINEERING AND TECHNOLOGY
SRM INSTITUTE OF SCIENCE AND TECHNOLOGY
KATTANKULATHUR-603 203
JANUARY 2022

AIM

To create a Calendar which allows user to view calendar in a neat and convenient manner.

ABSTRACT

This project is a simple project built in C language.
This project has following features –

1. It displays a nicely formatted calendar of every day of every month.
2. The calendar application presented here is a very simple console application developed using C programming language.
3. It is compiled in Eclipse using eclipse java compiler.

ALGORITHM

Step 1: Start.

Step 2: Declare int variable- get_1st_weekDay, year, day.

Step 3: Statement: day=Remainder of {[(year-1) × 365] + [(year-1)/4]– [{year-1)/100] + [(year/400)+1]} divided by 7.

Step 4: Declare int variables- year, month, day, daysInMonth, weekday, startingDay.

Step 5: Print “Enter your desired year:” in next line.

Step 6: Read the Year.

Step 7: To print months, we use pointer array.

Step 8: Declare array- monthDay.

Step 9: Using if condition to define the monthDay for month 1.

Step 10: Statement: startingDay=get_1st_weekDay(year).

Step 11: Initializing a “for” loop with regard to the month.

Step 12: Print the name of the month.

Step 13: Print the name of the days.

Step 14: Initializing another “for” loop with regard to the week.

Step 15: Initializing another “for” loop with regard to the day.

Step 16: Print the day.

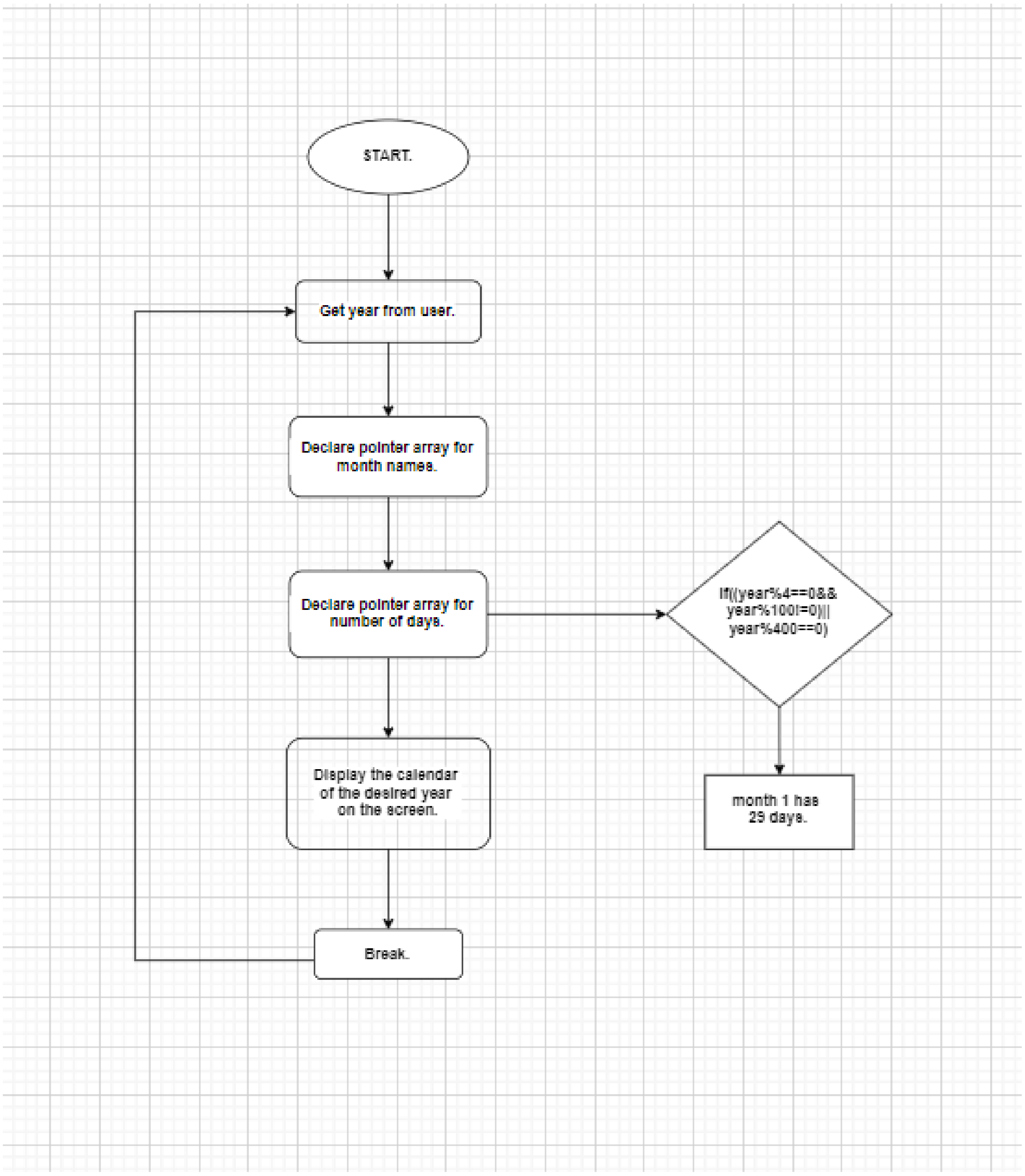
Step 17: Using if condition to set the format of the calender by aligning the dates and days in the proper order.

Step 18: Statement: startingDay=weekday.

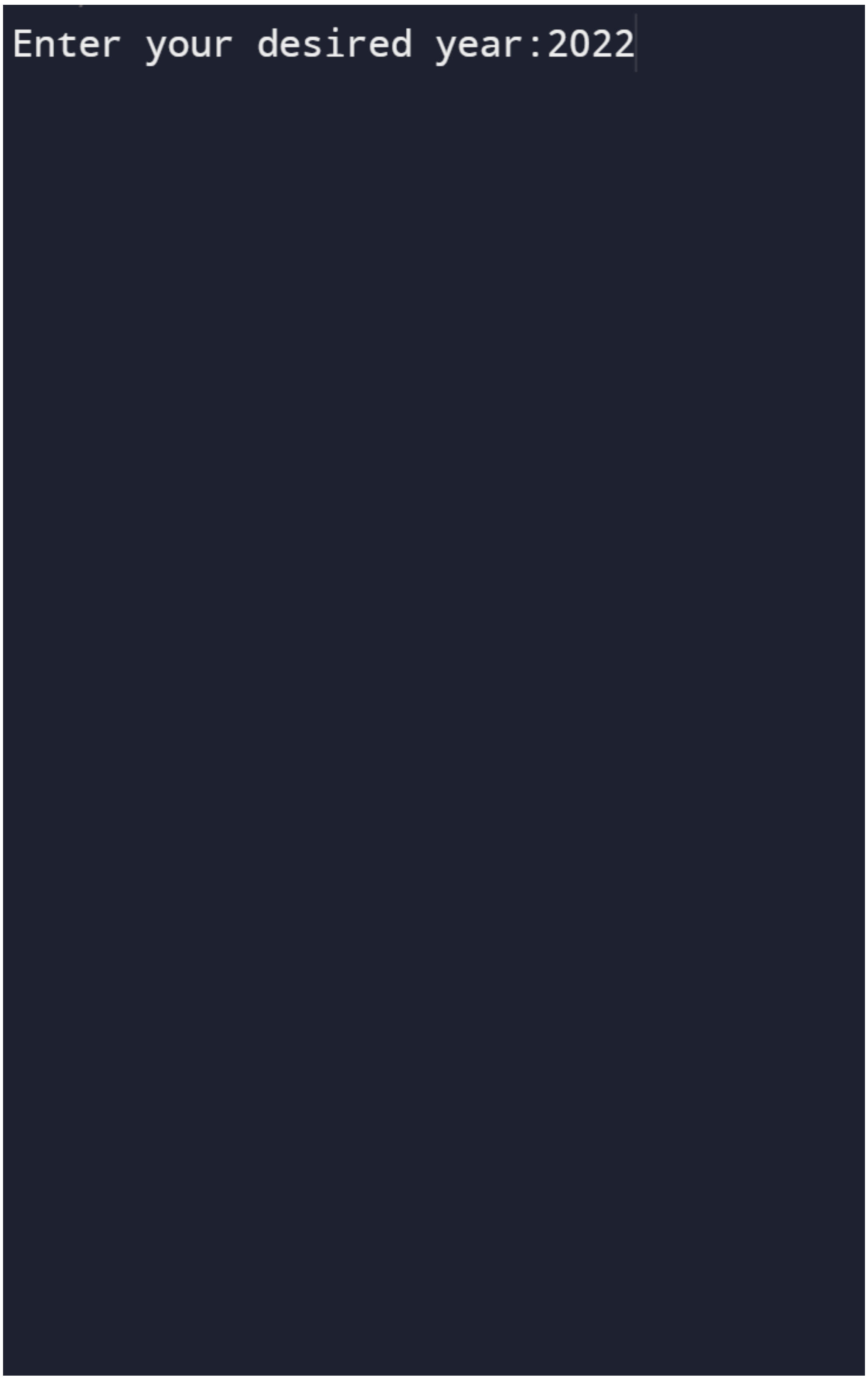
SOURCE CODE

```
#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;
}

int main()
{
   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","Oc
tober","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);

        if(++weekDay>6){
            printf("\n");
            weekDay=0;
        }
        startingDay=weekDay;
      }

   }
}
```

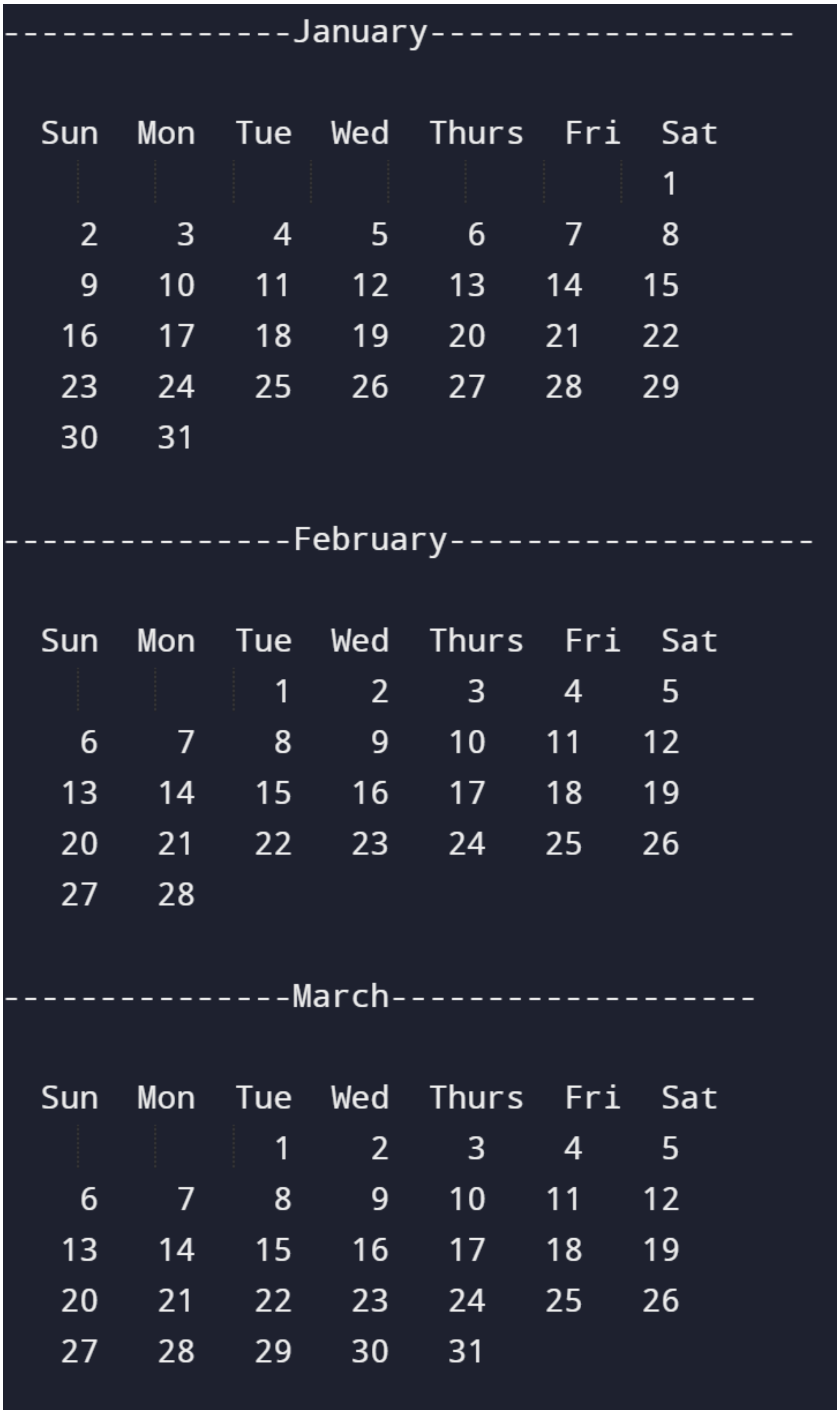
FLOWCHART


START.
Get year from user.
Declare pointer array for month names.
Declare pointer array for number of days.
If((year%4==0&& year%100!=0)|| year%400==0)
month 1 has 29 days.
Display the calendar of the desired year on the screen.
Break.

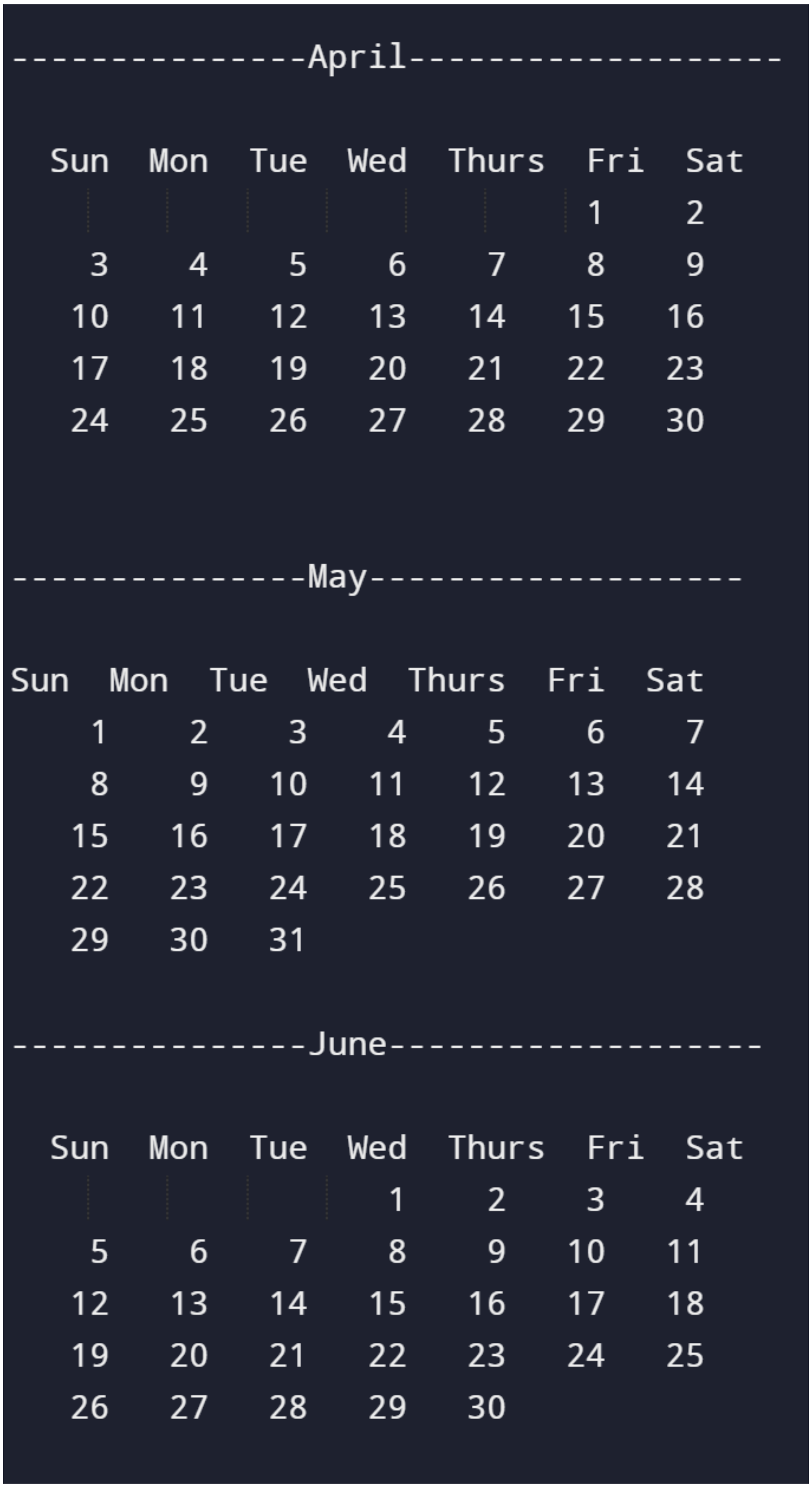
OUTPUT



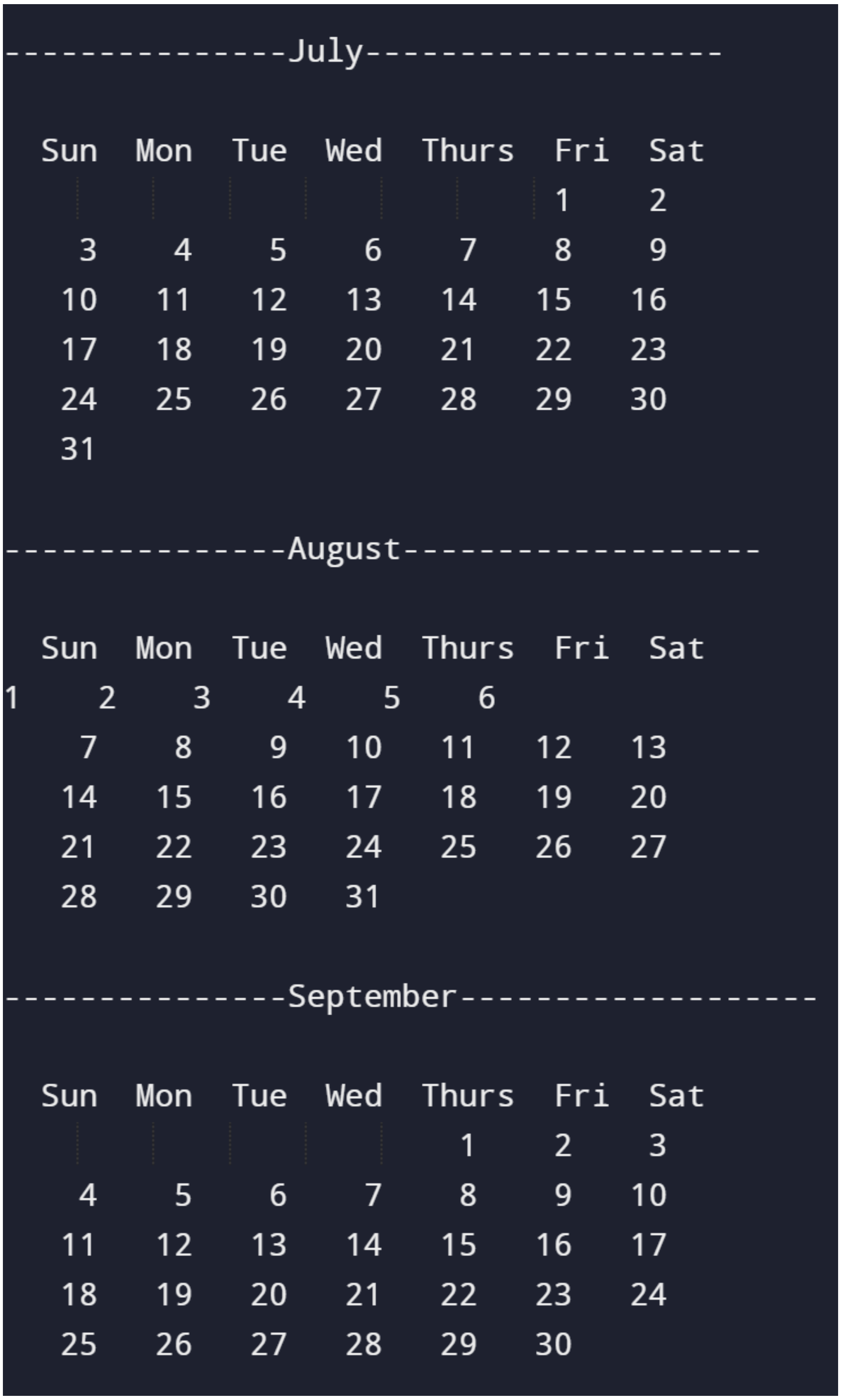
```
Enter your desired year:2022
```



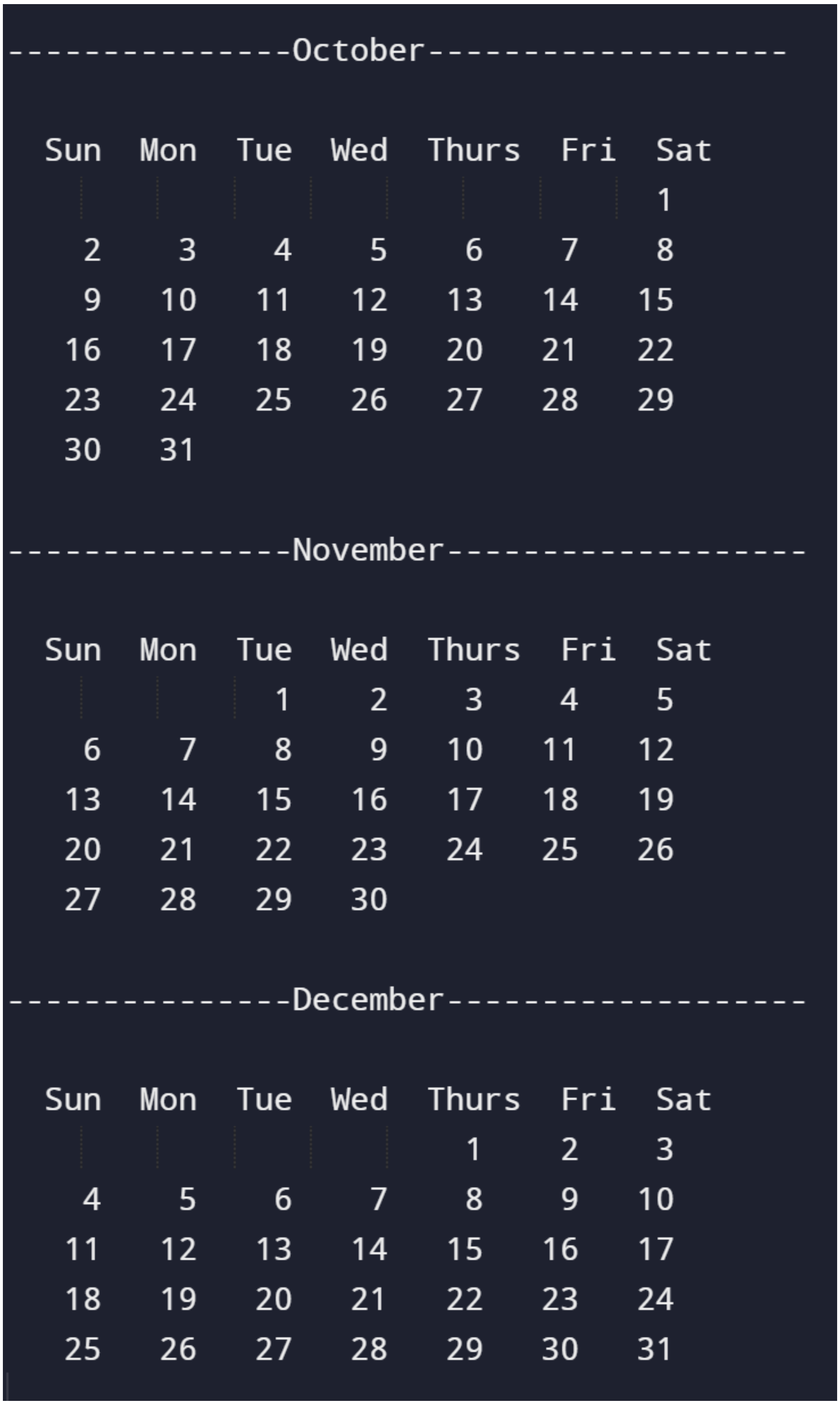
```
---------------January-------------------

  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

---------------February-------------------

  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

---------------March-------------------

  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
```



```
---------------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
```

RESULT

Our project calendar provides an easy access to a formatted calendar of any desired year in a neat and efficient manner.
Our project has succeeded in managing the data and providing the best output.

CONCLUSION

C is most useful for embedded systems, or applications that require the ability to be light-weight and have precise control over system resources. C is lacking a lot of the functionality that more contemporary languages feature but remains a core tool for Unix developers.

The two developers have tried their best to create a simple and optimized program that works as a calendar, with a user-friendly terminal for the executable file of the source code.