



**CHANDIGARH
UNIVERSITY**

Discover. Learn. Empower.

**NAAC
GRADE A+**

Accredited University

PROJECT FILE

C LANGUAGE CODE TO PRINT THE CALENDAR OF THE YEAR GIVEN BY THE USER

Name & UID:

RAKSHIT (24BCA10072)

SACHIN KUMAR SINGH (24BCA10079)

SHIVAM (24BCA10080)

Class/ Sec/ Group: 24 BCA 2 (A)

Subject: Computer Programming

Subject code: 24 CAH-101

Submitted to: Mr. Arvinder Singh Sir

Certificate

This is to certify that this project has been

made by

RAKSHIT (24BCA10072)

SACHIN KUMAR SINGH (24BCA10079)

SHIVAM (24BCA10080)

of

*branch **BCA** batch **2024***

*in the subject of **COMPUTER PROGRAMMING**,*

under my guidance and have been completed successfully.

Mr. Arvinder Singh Sir

ACKNOWLEDGEMENT

*I would like to express my special thanks of gratitude to my teacher **Mr. Arvinder Singh Sir**, who gave me the golden opportunity to this wonderful project/ case study in the subject of Digital Electronics.*

He also helped me in completing my project. I came to know about so many new things I am really thankful to them.

Secondly, I would also like to thank to my parents and friends who helped me a lot in finalizing this project within the limited time frame.

INDEX

❖ **INTRODUCTION OF C PROGRAMMING**

❖ **INTRODUCTION OF CALENDER**

❖ **FLOWCHART**

❖ **FLOWCHART EXPLANATION**

❖ **MAIN SOURCE CODE**

❖ **OUTPUT**

❖ **CODE EXPLANATION**

❖ **CONCLUSION**

INTRODUCTION OF C PROGRAM

C is a powerful and widely-used programming language that has been the foundation of many software systems since its development in the early 1970s by Dennis Ritchie at Bell Labs. It is a general-purpose, procedural language that provides low-level access to memory, making it ideal for system-level programming, such as operating systems, compilers, and embedded systems.

Key Features of C:

1. **Simple and Efficient:** C is known for its simplicity and efficiency, making it a popular choice for developing applications that require performance and control over system resources.
2. **Portable:** Programs written in C are highly portable, meaning they can be compiled and run on different computer systems with little or no modification.
3. **Modular Programming:** C supports modular programming, allowing developers to break down large programs into smaller, manageable functions or modules.
4. **Rich Standard Library:** C provides a vast collection of standard library functions for handling tasks such as input/output, string manipulation, memory management, and mathematical operations.
5. **Low-level Memory Access:** C allows direct manipulation of memory using pointers, giving developers fine-grained control over how data is stored and accessed.
6. **Flexibility:** C can be used for system-level programming as well as higher-level applications. It is versatile enough for tasks ranging from writing operating systems to developing game engines.

INTRODUCTION OF CALENDER

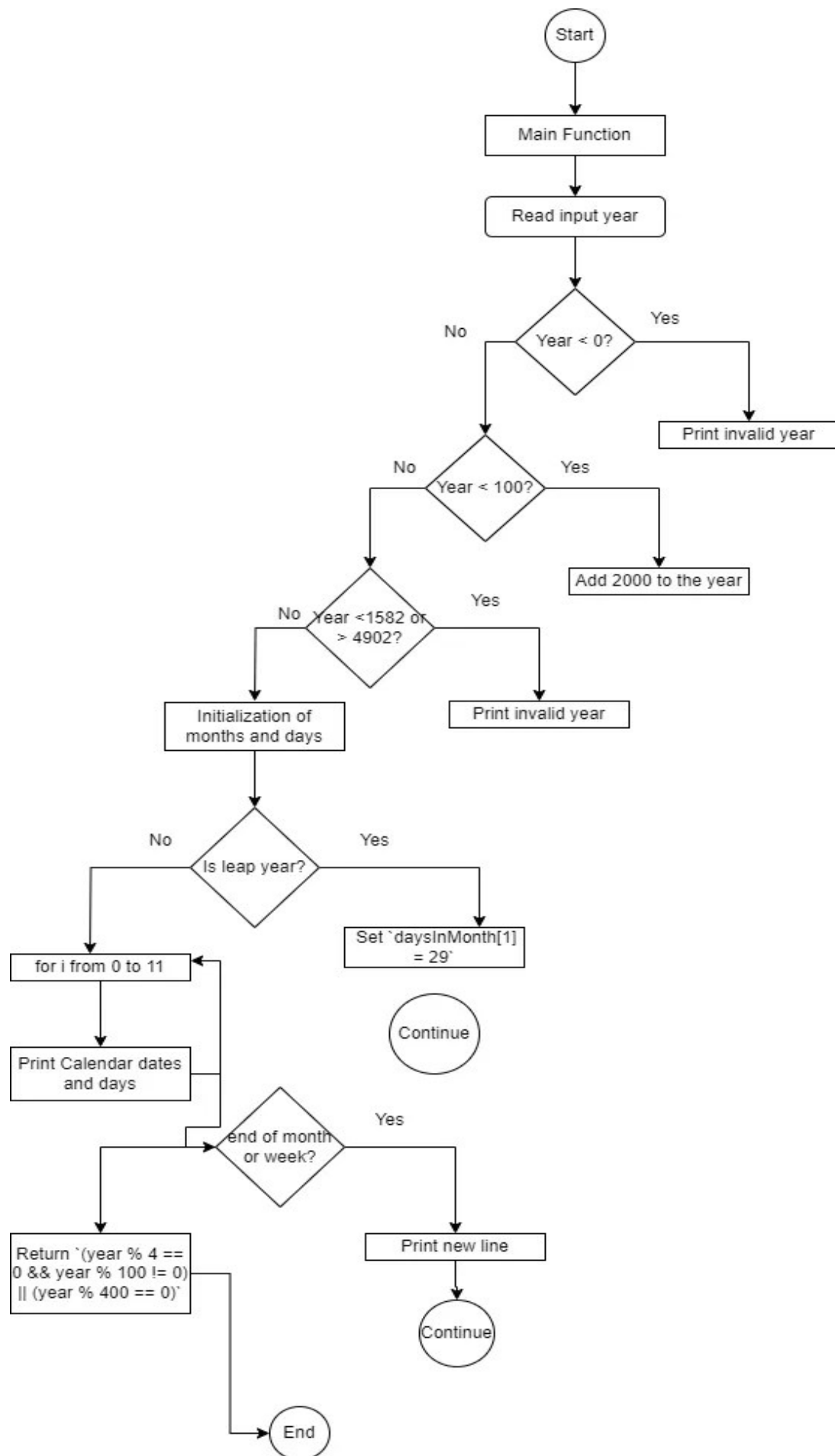
A calendar is a system of organizing units of time for the purpose of reckoning time over extended periods.

- A calendar is a useful tool that is used by everyone for their respective purpose.
- It is helpful in managing tasks, scheduling appointments, keeping track of your productivity etc.

How to Code A C Program To Display Complete Calendar (Example)?

Let us know how we can start with creating a calendar in the C programming language. To understand how it will work, i have added a flowchart for the same. It will help you get a proper idea of the calendar program. Take a look at it in the image below.

FlowChart



EXPLANATION

Here, we need to make an internationally accepted calendar like The Gregorian calendar. Therefore, we will need to add a few Lines to check if the user inputs are meeting all the conditions or not.

As shown in the flowchart, the main function will be executed first which will prompt for the user input. We will do input checking here to see if the entered year is valid or not. After this, we will call the function that we have created to display the calendar.

Let's see the implementation and explanation of the program below. First, we will write code and then we will try to understand every function properly.

SOURCE CODE

```
#include <stdio.h>

#include <stdbool.h>

bool isLeapYear(int year)
{
    return (year % 4 == 0 && year % 100 != 0) || (year % 400 == 0);
}

void displayCalendar(int year)
{
    const char *months[] =
    {
        "January", "February", "March", "April",
        "May", "June", "July", "August",
        "September", "October", "November", "December"
    };

    int daysInMonth[] = {31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31};
    if (isLeapYear(year))
    {
        daysInMonth[1] = 29; // February has 29 days in a leap year
    }

    printf("\nCalendar for year %d:\n", year);
    for (int month = 0; month < 12; ++month)
    {
        printf("\n\t%s:\n", months[month]);
        printf(" Sun Mon Tue Wed Thu Fri Sat\n");
    }
}
```

// Calculate the starting day of the month (0: Sunday, 1: Monday, ..., 6: Saturday)

int startingDay = year + (year - 1) / 4 - (year - 1) / 100 + (year - 1) / 400;

for (int i = 0; i < month; ++i)

{

startingDay += daysInMonth[i];

}

startingDay %= 7;

// Print the empty cells for the starting day

for (int i = 0; i < startingDay; ++i)

{

printf(" ");

}

// Print the days of the month

for (int day = 1; day <= daysInMonth[month]; ++day)

{

printf("%4d ", day);

if ((startingDay + day) % 7 == 0 || day == daysInMonth[month])

{

printf("\n");

}

}

}

```
}  
  
int main()  
{  
    int year;  
  
    printf("Enter a year to display the calendar(e.g. 2023): ");  
    scanf("%d", &year);  
  
    if (year < 0) {  
        printf("Invalid year!\n");  
        return 1;  
    }  
  
    if (year < 100)  
    {  
        year += 2000; // Assume years less than 100 are in the 2000s  
    }  
  
  
    // Range where gregorian calendar can be shown without accuracies and  
    overflows.  
  
    if (year < 1582 || year > 4902)  
    {  
        printf("Invalid year!\n");  
        return 1;  
    }  
  
    displayCalendar(year);  
    return 0;  
}
```

OUTPUT

Enter a year to display the calendar(e.g. 2024): 2024

Calendar for the year 2024:

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 29

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:

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

May:

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

June:

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

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 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

December:

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

EXPLANATION

Let's break down the code into simple steps:

Step 1:

Include Headers

The code starts by including two important header files : `stdio.h` for input and output functions, and `stdbool.h` for using the `bool` data type and `true/false` values.

Step 2 :

Define `isLeapYear` function

This function helps to find out If a given year is a leap year. Leap years have an extra day (February29). Strictly speaking, we will have to not just see the divisibility by 4 but also by 400.

It does this by using the modulo operator `%` to check if the year is divisible by 4 and not divisible by 100, or if it's divisible by 400. If conditions are met, it return `true` or `false`.

According to the Gregorian calendar rules:

- Years divisible by 4 are potential leap years.
- Years divisible by 100 are not leap years unless they are also divisible by 400, in which case they are leap years.

Step 3:

Define `displayCalendar` function

The next function is an important one! This is the function where we pass the input (int year) as the parameter. You can make a separate function calendar year for taking the input if you want.

This function displays a calendar for the specified year by user. It initializes “months” and “daysInMonth” for storing month names and the no of days in each month respectively.

If the year is a leap year, then we must add one more day in one month. So, February will have 29 days for this case.

The function shows the calendar header with the year. For each of the 12 months, it prints the month’s name, day headers, calculates the starting day, aligns days, and handles line breaks.

Here, we have made two arrays, one stores the number of days in one month, and the other stores the name of the months. We need to output the right weekday at the correct position for all the dates.

Step 4:

Define the main Function

- It prompts the user to enter a year and stores it in the `year` variable. It checks if the entered year is negative. If so, it prints an error message and returns 1 to indicate an error.
- If the entered year is less than 100, it assumes that two-digit years refer to the 2000s and adjusts the year accordingly.

- It checks if the entered year falls within a range (1582 to 4902) where the [Gregorian calendar](#) is valid.

If not, it prints an error message and returns 1.

- Finally, it calls the `displayCalendar` function to show the calendar for the specified year.

This is how we can create our program for displaying all the days in the calendar.

CONCLUSION

So, this is how you can print or display month-by-month for a calendar year in C programming. I hope that you have understood the code and the functions in it with the help of the explanation provided.

The calendar application has a lot of use in other software applications as well. It is a basic tool that you can include in your projects to make them a little more elaborate. For more such programs on programming, check out our other articles as well!

Bibliography:

www.crome.com

www.geeksforgeeks.com