Calendar Generator in C - Project Report

Name :- Divyanshu Raman Pandey

Sap Id :- 590026079

Course :- Computer science and enginerring

Batch :- 20

Year :- 2025-2026

Subject :- C language

Submitted to : Rahul pasad

1. Abstract

This project focuses on developing a C program that generates a monthly calendar for any user-entered month and year. The program uses fundamental concepts of C such as functions, loops, conditional statements, and arrays. To ensure accurate day alignment, the program implements Zeller's Congruence, a mathematical formula used to determine the day of the week for any given date. A leap year detection mechanism is included to correctly calculate February's days.

The solution is designed using a modular structure with functions for leap year checking, day calculation, month-day retrieval, holiday listing, and calendar printing. The program also displays important holidays for the selected month, making the output more informative and practical. Testing was performed across multiple months, including leap years, to verify accuracy.

Overall, the project demonstrates how mathematical algorithms and structured programming techniques can be combined to build an efficient, user-friendly calendar generator using C.

2. Problem Definition

The problem addressed in this project is to design and implement a C program capable of generating an accurate monthly calendar based on the user's input of month and year. The program must correctly determine the number of days in each month, identify leap years, and calculate the starting weekday using a reliable algorithm. Additionally, the system should display important holidays for the selected month to make the output more informative.

The goal is to create a simple, user-friendly calendar generator that demonstrates the practical application of C programming concepts such as modular design, conditional statements, switch-case structures, and mathematical logic. The solution should ensure correctness, readability, and maintainability while providing a meaningful real-world use case for date computation in C.

3. System Design

3.1 Flow of the Program

1. User inputs month and year.

2. Program checks if the given year is a leap year.

3. Zeller's Congruence is applied to compute the day of the week.

4. The calendar is printed in a structured grid format.

5. The list of holidays for that specific month is displayed.

3.2 Algorithms Used

Leap Year Check Algorithm:

- If a year is divisible by 4, it may be a leap year.

- If divisible by 100, it must also be divisible by 400.

- Otherwise, it is not a leap year.

Day of Week Calculation (Zeller's Congruence):

A standard formula used to compute weekday for any given date

4. Implementation Details

The program is implemented in C using a modular structure, where each major task is handled by a separate function. This improves readability, maintainability, and reusability of the code.

1. *isLeapYear()*
   This function checks whether the given year is a leap year using standard divisibility rules. It helps in deciding whether February has 28 or 29 days.
2. *getDaysInMonth()*
   A switch-case structure is used to return the number of days in the selected month. For February, it depends on the leap year result.
3. *dayOfWeek() – Zeller's Congruence*
   This function calculates the weekday of the first day of the given month and year. It adjusts months and years as required by Zeller's formula and returns a value from 0–6 representing Sunday–Saturday.
4. *printHolidays()*
   This function contains predefined holidays for all months. Depending on the selected month, it prints the list of important events and festivals.
5. *printCalendar()*
   Using the starting weekday and total days in the month, this function prints the calendar in a structured grid

format. It aligns the dates properly under the days of the week.

6. *main() Function*
   The program begins by asking the user for the month and year. It then calls the printCalendar() function, which internally uses the other functions to generate the final output.

Overall, the implementation showcases proper use of functions, conditionals, loops, and mathematical logic to build a fully functional calendar generator in C.

5. Testing & Results

To verify the correctness and reliability of the program, multiple test cases were executed using different combinations of months and years. The testing focused on checking the accuracy of leap year calculations, correct weekday alignment, month-day counts, and the proper display of holidays.

Test Case 1: January 2025

- *Input: 01 2025*
- *Output: Correctly displayed calendar with January starting on Wednesday.*
- *Holidays such as New Year, National Youth Day, and Republic Day appeared correctly.*

Test Case 2: February 2024 (Leap Year)

- *Input: 02 2024*
- *Output: Calendar showed 29 days, confirming correct leap year logic.*
- *February started on Thursday, matching real-world data.*

Test Case 3: November 2025

- *Input: 11 2025*
- *Output: Calendar formatting was correct, holidays like Diwali, Children's Day, and Constitution Day displayed properly.*

Test Case 4: Invalid Month

- *Input: 13 2024*
- *Output: The program gracefully handled the invalid month by returning zero days, preventing crashes or undefined behavior.*

Overall Results

- *Calendar alignment was accurate for all tested inputs.*
- *Zeller's Congruence produced correct weekdays.*
- *Holiday list matched the corresponding month.*
- *The program executed without errors or segmentation faults.*

The testing confirms that the calendar generator is reliable, accurate, and meets the expected functional requirements

7. Conclusion & Future Work

Conclusion

The Calendar Generator project successfully demonstrates how core concepts of C programming can be combined to create a functional and user-friendly application. By using modular functions, leap year validation, and Zeller's Congruence for day calculation, the program is able to produce accurate monthly calendars for any given month and year. The inclusion of a holiday list makes the output more informative and practical. This project also enhanced understanding of algorithms, conditional logic, and structured programming—skills that are essential for building more advanced applications.

Future Work

The current version of the program can be expanded in several ways to make it more powerful and interactive. Possible improvements include:

- *Adding an option to generate a full yearly calendar.*
- *Allowing users to add custom holidays or events.*
- *Highlighting weekends and holidays within the printed calendar.*
- *Improving the interface with color-coded output or a simple GUI.*

- *Integrating system date/time to show the current month automatically.*

These enhancements would further increase the usefulness of the application and provide opportunities to explore new programming features and algorithms.

7. References

1. *Zeller, C. – Zeller's Congruence Formula (Standard algorithm for calculating day of the week).*
2. *Kernighan, B. W., & Ritchie, D. M. – The C Programming Language, Prentice Hall.*
3. *GeeksforGeeks – Articles on C functions, switch cases, and leap year logic.*
4. *TutorialsPoint – C Programming Documentation – Standard references for syntax and library functions.*
5. *Online Calendar Verification Tools – Used to validate correctness of generated calendars.*

8. Code and input

```
#include <stdio.h>

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

int getDaysInMonth(int month, int year) {
    switch(month) {
        case 1: return 31;
        case 2: return isLeapYear(year) ? 29 : 28;
        case 3: return 31;
        case 4: return 30;
        case 5: return 31;
        case 6: return 30;
        case 7: return 31;
        case 8: return 31;
        case 9: return 30;
        case 10: return 31;
        case 11: return 30;
        case 12: return 31;
        default: return 0;
    }
}

// Zeller's Congruence
int dayOfWeek(int day, int month, int year) {
    if (month < 3) {
        month += 12;
        year -= 1;
    }
    int K = year % 100;
    int J = year / 100;
    int h = (day + (13 * (month + 1)) / 5 + K + K/4 + J/4 + 5*J) % 7;
    return (h + 6) % 7; // Convert to 0=Sun
}

void printHolidays(int month) {
    printf("\nHolidays:\n");

    switch(month) {

        case 1:
            printf("  1  - New Year\n");
```

```
52.             printf("  4  - World Braille Day\n");
53.             printf(" 12 - National Youth Day\n");
54.             printf(" 12 - Divyanshu's Birthday\n");
55.             printf(" 14 - Makar Sankranti (Approx)\n");
56.             printf(" 15 - Pongal (Approx)\n");
57.             printf(" 16 - Guru Gobind Singh Jayanti (Approx)\n");
58.             printf(" 23 - Netaji Subhas Chandra Bose Jayanti\n");
59.             printf(" 26 - Republic Day\n");
60.             printf(" 30 - Martyrs' Day\n");
61.             break;
62.
63.         case 2:
64.             printf("  1  - Indian Coast Guard Day\n");
65.             printf("  4  - World Cancer Day\n");
66.             printf("  7  - Rose Day\n");
67.             printf("  8  - Propose Day\n");
68.             printf("  9  - Chocolate Day\n");
69.             printf(" 10 - Teddy Day\n");
70.             printf(" 11 - Promise Day\n");
71.             printf(" 12 - Hug Day\n");
72.             printf(" 13 - Kiss Day\n");
73.             printf(" 14 - Valentine's Day\n");
74.             printf(" 19 - Shivaji Maharaj Jayanti\n");
75.             printf(" 20 - World Day of Social Justice\n");
76.             printf(" 27 - World NGO Day\n");
77.             printf(" 28 - National Science Day\n");
78.             break;
79.
80.         case 3:
81.             printf("  1  - Zero Discrimination Day\n");
82.             printf("  3  - World Wildlife Day\n");
83.             printf("  4  - National Safety Day\n");
84.             printf("  8  - Women's Day\n");
85.             printf(" 10 - Maha Shivaratri (Approx)\n");
86.             printf(" 15 - Consumer Rights Day\n");
87.             printf(" 20 - Happiness Day\n");
88.             printf(" 21 - Holi (Approx)\n");
89.             printf(" 22 - World Water Day\n");
90.             printf(" 23 - Shaheed Diwas\n");
91.             printf(" 27 - World Theatre Day\n");
92.             printf(" 31 - Easter (Approx)\n");
93.             break;
94.
95.         case 4:
96.             printf("  1  - April Fool's Day\n");
97.             printf("  2  - Autism Awareness Day\n");
98.             printf("  7  - World Health Day\n");
99.             printf(" 10 - Eid al-Fitr (Approx)\n");
```

```
100.                 printf(" 13 - Jallianwala Bagh Remembrance\n");
101.                 printf(" 14 - Ambedkar Jayanti\n");
102.                 printf(" 15 - Vishu / Baisakhi (Approx)\n");
103.                 printf(" 17 - Ram Navami (Approx)\n");
104.                 printf(" 18 - World Heritage Day\n");
105.                 printf(" 22 - Earth Day\n");
106.                 printf(" 23 - World Book Day\n");
107.                 printf(" 29 - Dance Day\n");
108.                 break;
109.
110.             case 5:
111.                 printf("  1  - Labour Day\n");
112.                 printf("  3  - Press Freedom Day\n");
113.                 printf("  8  - World Red Cross Day\n");
114.                 printf(" 12 - Nurses Day\n");
115.                 printf(" 15 - Families Day\n");
116.                 printf(" 17 - Telecom Day\n");
117.                 printf(" 21 - Anti-Terrorism Day\n");
118.                 printf(" 22 - Biodiversity Day\n");
119.                 printf(" 24 - Buddha Purnima (Approx)\n");
120.                 printf(" 31 - No Tobacco Day\n");
121.                 break;
122.
123.             case 6:
124.                 printf("  1  - Parents Day\n");
125.                 printf("  5  - Environment Day\n");
126.                 printf("  8  - Oceans Day\n");
127.                 printf(" 12 - Child Labour Day\n");
128.                 printf(" 14 - Blood Donor Day\n");
129.                 printf(" 18 - Father's Day (Approx)\n");
130.                 printf(" 19 - Reading Day\n");
131.                 printf(" 20 - Refugee Day\n");
132.                 printf(" 21 - Yoga Day\n");
133.                 printf(" 23 - Olympic Day\n");
134.                 printf(" 29 - Statistics Day\n");
135.                 break;
136.
137.             case 7:
138.                 printf("  1  - Doctor's Day\n");
139.                 printf("  4  - USA Independence Day\n");
140.                 printf("  7  - Chocolate Day\n");
141.                 printf(" 11 - Population Day\n");
142.                 printf(" 17 - Justice Day\n");
143.                 printf(" 20 - Muharram (Approx)\n");
144.                 printf(" 22 - Pi Approximation Day\n");
145.                 printf(" 26 - Kargil Vijay Diwas\n");
146.                 printf(" 28 - Nature Conservation Day\n");
147.                 printf(" 29 - Tiger Day\n");
```

```
148.                 break;
149.
150.             case 8:
151.                 printf("  6  - Hiroshima Day\n");
152.                 printf("  7  - Handloom Day\n");
153.                 printf("  9  - Quit India Movement Day\n");
154.                 printf(" 11 - Raksha Bandhan (Approx)\n");
155.                 printf(" 12 - Youth Day\n");
156.                 printf(" 15 - Independence Day\n");
157.                 printf(" 19 - Photography Day\n");
158.                 printf(" 19 - Janmashtami (Approx)\n");
159.                 printf(" 23 - Slave Trade Remembrance\n");
160.                 printf(" 29 - Sports Day\n");
161.                 printf(" 30 - Small Industry Day\n");
162.                 break;
163.
164.             case 9:
165.                 printf("  5  - Teacher's Day\n");
166.                 printf("  7  - Janmashtami (Approx)\n");
167.                 printf("  8  - Literacy Day\n");
168.                 printf(" 14 - Hindi Diwas\n");
169.                 printf(" 15 - Engineer's Day\n");
170.                 printf(" 16 - Ozone Day\n");
171.                 printf(" 21 - Peace Day\n");
172.                 printf(" 22 - World Rose Day\n");
173.                 printf(" 23 - Autumn Equinox (Approx)\n");
174.                 printf(" 27 - Tourism Day\n");
175.                 printf(" 29 - World Heart Day\n");
176.                 break;
177.
178.             case 10:
179.                 printf("  1  - Older Persons Day\n");
180.                 printf("  2  - Gandhi Jayanti\n");
181.                 printf("  4  - Animal Day\n");
182.                 printf("  8  - Air Force Day\n");
183.                 printf(" 12 - Dussehra (Approx)\n");
184.                 printf(" 13 - Calamity Control Day\n");
185.                 printf(" 16 - Food Day\n");
186.                 printf(" 17 - Poverty Eradication Day\n");
187.                 printf(" 24 - United Nations Day\n");
188.                 printf(" 31 - Patel Jayanti\n");
189.                 printf(" 31 - Halloween\n");
190.                 break;
191.
192.             case 11:
193.                 printf("  1  - Diwali (Approx)\n");
194.                 printf("  2  - Govardhan Puja (Approx)\n");
195.                 printf("  3  - Bhai Dooj (Approx)\n");
```

```
                printf("  6  - Chhath Puja Begins (Approx)\n");
                printf("  9  - Chhath Puja Ends (Approx)\n");
                printf(" 12 - Guru Nanak Jayanti\n");
                printf(" 14 - Children's Day\n");
                printf(" 19 - National Integration Day\n");
                printf(" 26 - Constitution Day\n");
                break;

            case 12:
                printf("  1  - AIDS Day\n");
                printf("  3  - Disability Day\n");
                printf("  4  - Navy Day\n");
                printf("  7  - Armed Forces Flag Day\n");
                printf(" 10 - Human Rights Day\n");
                printf(" 22 - Mathematics Day\n");
                printf(" 24 - Consumer Rights Day\n");
                printf(" 25 - Christmas\n");
                printf(" 31 - New Year's Eve\n");
                break;
        }
    }

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

        int days = getDaysInMonth(month, year);
        int startDay = dayOfWeek(1, month, year);

        printf("\n  %s %d\n", months[month], year);
        printf("Su Mo Tu We Th Fr Sa\n");

        for (int i = 0; i < startDay; i++)
            printf("   ");

        for (int d = 1; d <= days; d++) {
            printf("%2d ", d);
            if ((d + startDay) % 7 == 0)
                printf("\n");
        }

        printf("\n");
        printHolidays(month);
    }
```

```
242.
243.      int main() {
244.          int month, year;
245.
246.          printf("Enter month and year (e.g., 11 2025): ");
247.          scanf("%d %d", &month, &year);
248.
249.          printCalendar(month, year);
250.
251.          return 0;
252.      }
```

9. Output

Output Clear

```
Enter month and year (e.g., 11 2025): 10 2000

  October 2000
Su Mo Tu We Th Fr Sa
 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

Holidays:
  1  - Older Persons Day
  2  - Gandhi Jayanti
  4  - Animal Day
  8  - Air Force Day
 12 - Dussehra (Approx)
 13 - Calamity Control Day
 16 - Food Day
 17 - Poverty Eradication Day
 24 - United Nations Day
 31 - Patel Jayanti
 31 - Halloween
```

Output Clear

```
Enter month and year (e.g., 11 2025): 11 2020

  November 2020
Su Mo Tu We Th Fr Sa
 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

Holidays:
  1  - Diwali (Approx)
  2  - Govardhan Puja (Approx)
  3  - Bhai Dooj (Approx)
  6  - Chhath Puja Begins (Approx)
  9  - Chhath Puja Ends (Approx)
 12 - Guru Nanak Jayanti
 14 - Children's Day
 19 - National Integration Day
 26 - Constitution Day


=== Code Execution Successful ===
```