A Project Report

On

“Calendar”

For The Course

“Software Development Project-I”

By

Md. Asadul Islam (IT-20049)

Emu Bhuiyan (IT-20051)

Supervised by

S.M. Shamim

Assistant Professor

Department of ICT, MBSTU.

DEPARTMENT OF INFORMATION AND COMMUNICATION TECHNOLOGY

MAWLANA BHASHANI SCIENCE AND TECHNOLOGY UNIVERSITY

SANTOSH, TANGAIL-1902, Dhaka, Bangladesh

Declaration

This is to certify that the work presented in this project is carried out by the candidate under the supervision of **S.M. Shamim** in the department of Information and Communication Technology, MBSTU, Tangail, Bangladesh. It is also declared that neither of this project has been submitted anywhere else for any degree or diploma. Information derived from the published and unpublished work of others has been acknowledged in the text and a list of references is given.

Signature of Supervisor

S.M Shamim

Assistant Professor

Dept. of ICT,MBSTU

Acknowledgements

I must sense grateful to the Almighty Allah to complete the dissertation. At the outset, I would like to express gratitude to my supervisor S.M Shamim, Assistant Professor , Dept. of Information and Communication Technology, MBSTU who has supported our plan to continue Software Development Project-I. I also like to express gratitude to our supervisor, for his valuable guidance and insight, encouragement , support and reliance throughout the project. However it is not possible acknowledge properly the effort of our honorable teacher in writing words.

We are, as always , indebted to our family. The love and support of our parents remain bedrock of our life.

Contents

Page No.

Chapter – 1

Introduction of Project

This project on Calendar in C programming language is a console application without graphics. To make the calendar look colorful, many windows properties have been used in this project. Besides the color used in backgrounds, for front page of console we have used violet color and there are so many color have been used in every step of console output.

In this project, you can find out the day corresponding to a given date and view the days and dates corresponding to a particular month + year, adding and view note on a particular date. The source code is not that long, over 360 lines. It is compiled in Code::Blocks IDE with GCC compiler.

To help you understand the calendar project better, there are lots of comments within the source code. This project is aimed to show you

“How to make English & Bangla Calendar using C Programming Language.”

Chapter 2

2.1 - Programming Language

Programming Language : As we know, to communicate with a person, we need a specific language, similarly to communicate with computers, programmers also need a language is called Programming language.

Before learning the programming language, let's understand what is language?

What is Language?

Language is a mode of communication that is used to share ideas, opinions with each other. For example, if we want to teach someone, we need a language that is understandable by both communicators.

What is a Programming Language?

A programming language is a computer language that is used by programmers (developers) to communicate with computers. It is a set of instructions written in any specific language (C, C++, Java, Python) to perform a specific task.

A programming language is mainly used to develop desktop applications, websites, and mobile applications.

Types of programming language:

1. Low-level programming language:

Low-level language is machine-dependent (0s and 1s) programming language. The processor runs low- level programs directly without the need of a compiler or interpreter, so the programs written in low-level language can be run very fast.

Low-level language is further divided into two parts -

i. Machine Language:

Machine language is a type of low-level programming language. It is also called as machine code or object code. Machine language is easier to read because it is normally displayed in binary or hexadecimal form (base 16) form. It does not require

a translator to convert the programs because computers directly understand the machine language programs.

The advantage of machine language is that it helps the programmer to execute the programs faster than the high-level programming language.

ii. Assembly Language:

Assembly language (ASM) is also a type of low-level programming language that is designed for specific processors. It represents the set of instructions in a symbolic and human-understandable form. It uses an assembler to convert the assembly language to machine language.

The advantage of assembly language is that it requires less memory and less execution time to execute a program.

2. High-level programming language:

High-level programming language (HLL) is designed for developing user-friendly software programs and websites. This programming language requires a compiler or interpreter to translate the program into machine language (execute the program).

The main advantage of a high-level language is that it is easy to read, write, and maintain.

High-level programming language includes Python, Java, JavaScript, PHP, C#, C++, Objective C, Cobol, Perl, Pascal, LISP, FORTRAN, and Swift programming language.

A high-level language is further divided into three parts -

i. Procedural Oriented programming language:

Procedural Oriented Programming (POP) language is derived from structured programming and based upon the procedure call concept. It divides a program into small procedures called routines or functions.

Procedural Oriented programming language is used by a software programmer to create a program that can be accomplished by using a programming editor like IDE, Adobe Dreamweaver, or Microsoft Visual Studio.

The advantage of POP language is that it helps programmers to easily track the program flow and code can be reused in different parts of the program.

Example: C, FORTRAN, Basic, Pascal, etc.

03.Middle-level programming language:

Middle-level programming language lies between the low-level programming language and high-level programming language. It is also known as the intermediate programming language and pseudo-language.

A middle-level programming language's advantages are that it supports the features of high-level programming, it is a user-friendly language, and closely related to machine language and human language.

Example: C, C++, language

2.2 - C Programming Language

C is a popular, simple, and flexible general-purpose computer programming language. Dennis M Ritchie develops it in 1972 at AT&T. It is a combination of both low-level programming language as well as a high-level programming language. It is used to design applications like Text Editors, Compilers, Network devices, and many more**.**

Most Important Features of C Language:

- Simple and Efficient.
- Fast.
- Portability.
- Extensibility.
- Function-Rich Libraries.
- Dynamic Memory Management.
- Modularity With Structured Language.

Advantages:

- C language is easy to learn.
- It is fast, efficient, portable, easy to extend, powerful, and flexible programming language.
- It is used to perform complex calculations and operations such as MATLAB.
- It provides dynamic memory allocation to allocate memory at the run time.

Disadvantages:

- In the C programming language, it is very difficult to find the errors.
- C does not support the concepts of constructors, destructors, abstraction, polymorphism, encapsulation, and namespace like OOPs.

2.3 IDE details

Code::Blocks:

Code::Blocks is a free, open-source cross-platform IDE that supports multiple compilers including GCC, Clang and Visual C++. It is developed in C++ using wxWidgets as the GUI toolkit. Using a plugin architecture, its capabilities and features are defined by the provided plugins. Currently, Code::Blocks is oriented towards C, C++, and Fortran. It has a custom build system and optional Make support.

Features

Compilers

Code::Blocks supports multiple compilers, including GCC, MinGW, Digital Mars, Microsoft Visual C++, Borland C++, LLVM Clang, Watcom, LCC and the Intel C++ compiler. Although the IDE was designed for the C++ language, there is some support for other languages, including Fortran and D. A plug-in system is included to support other programming languages.

Code editor

The IDE features syntax highlighting and code folding (through its Scintilla editor component), C++ code completion, class browser, a hex editor and many other utilities. Opened files are organized into tabs. The code editor supports font and font size selection and personalized syntax highlighting colors.

Debugger

The Code::Blocks debugger has full breakpoint support. It also allows the user to debug their program by having access to the local function symbol and argument display, user-defined watches, call stack, disassembly, custom memory dump, thread switching, CPU registers and GNU Debugger Interface.

Chapter – 3

Code Header file:

- #include<stdio.h>
- #include<stdlib.h>
- #include<windows.h>
- #include<string.h>

#include<stdio.h>:

stdio.h is a header file which has the necessary information to include the input/output related functions in our program. Example printf, scanf etc.

#include<stdlib.h>:

stdlib.h is the header of the general purpose standard library of C programming language which includes functions involving memory allocation, process control, conversions and others. It is compatible with C++ and is known as ctdlib in C++. The name "stdlib" stands for "standard library".

#include<windows.h>:

windows.h is a Windows-specific header file for the C and C++ programming languages which contains declarations for all of the functions in the Windows API, all the common macros used by Windows programmers.

#include<string.h>:

The string.h header defines one variable type, one macro, and various functions for manipulating arrays of characters.

Chapter – 4

4.1 - Function Name & Details

firstDayOfTheYear : (This function is used for finding the first day of the English year & this function return a integer value)

banglaFirstDayOfTheYear : (This function is used for finding the first day of the Bangla year & this function return a integer value)

checkingLeapYear: (This function is used for checking the specific English year is leap year or not .The return type of this function is void)

printingYear: (This function is used for printing English year & it's return type is void)

printingMonth: (This function is used for printing the English month name and return type of this function is void)

PrintingBanglaYear: (This function is used for printing Bangla year & it's return type is void)

PrintingBanglaMonth: (This function is used for printing the Bangla month name and return type of this function is void)

PrintingSpecificDate: (This function is used for both English and Bangla Calendar for printing specific date and identifies the weekdays of that date and return type is void)

AddNote: (This function is used for adding note both for English and Bangla Calendar & return type is void)

ViewNote: (This function is used for viewing the added note.The return type of this function is void.)

Chapter – 5

5.1 – Source Code

Source Code:

```
#include <stdio.h>

#include <stdlib.h>C

#include<windows.h>

#include<string.h>

int year,month,day,daysInMonth,weekDay=0,startingDay;

char *months[]=
{"January","February","March","April","May","June","July","August","
September","October","November","December"};

int monthDay[]= {31,28,31,30,31,30,31,31,30,31,30,31};

char *monthsBangla[]=
{"Baishakh","Jaishtha","Ashar","Srabon","Bhadra","Ashwin","Karttik",
"Agrahayan","Paush","Magh","Falgun","Chaitra"};

int monthDayBangla[]= {31,31,31,31,31,30,30,30,30,30,30,30};
```

```
//Finding English 1st Date of the Year
int firstDayOfTheYear(int year){
    int d;
    d = (((year - 1) * 365) + ((year - 1) / 4) - ((year - 1) / 100) + ((year) / 400) + 1) % 7;
    return d;
}
//Finding Bangla 1st Date of the Year
int banglaFirstDayOfTheYear(int year)
{
    int d;
    year = year+593;
    d = (14 + 26 * (4+ 1) / 10 + year + year / 4-year /100 + year /400 ) % 7 ;
    return d;
}
//Cheaking Leapyear
void cheakingLeapYear(){
    if((year%4==0&&year%100!=0)||year%400==0)
        monthDay[1]=29;
}
```

```
//Cheaking Bangla Leapyear
/*void cheakingBanglaLeapYear()
{
   if(((year+594)%4==0&&(year+594)%100!=0)||(year+594)%400==0)
      monthDayBangla[10]=30;
}
*/
//Function For Printing English Year

void PrintingYear()
{
   system("cls");
   cheakingLeapYear();
   startingDay=firstDayOfTheYear(year);

   for(month=0; month<12; month++)
   {

      daysInMonth=monthDay[month];
```

```
    printf("\n\n---------------%s------------------\n",months[month]);
    printf("\n  Sun  Mon  Tue  Wed  Thu  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;
    }
  }

}
```

```
//Function For Printing English Month
void PrintingMonth(int month)
{
    system("cls");
    cheakingLeapYear();
    startingDay=firstDayOfTheYear(year);

    daysInMonth=monthDay[month];
    printf("\n\n--------------%s------------------\n",months[month]);
    printf("\n  Sun  Mon  Tue  Wed  Thu  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;
    }
}
//Printing Bangla Year
void PrintingBanglaYear()
{
    system("cls");
  //if(((year+594)%4==0&&(year+594)%100!=0)||(year+594)%400==0)
        //monthDayBangla[10]=30;

    startingDay=banglaFirstDayOfTheYear(year);

    for(month=0; month<12; month++)
    {
        daysInMonth=monthDayBangla[month];
        printf("\n\n----------%s----------
\n",monthsBangla[month]);
```

```
    printf("\n Sat Sun Mon Tue Wed Thu Fri \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;
    }

  }
}
```

```
//Printing bangla Month
void PrintingBanglaMonth(int month){
    system("cls");
    //cheakingBanglaLeapYear();
    startingDay=banglaFirstDayOfTheYear(year);
    daysInMonth=monthDayBangla[month];
    printf("\n\n--------------%s------------------\n",monthsBangla[month]);
    printf("\n Sat  Sun  Mon  Tue  Wed  Thu  Fri \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;
    }
}
```

```
//Printing Specific Date
void PrintingSpecificDate(int d,int m,int y)
{
    int h;
    if (m == 1)
    {
        m = 13;
        y--;
    }
    if (m == 2)
    {
        m = 14;
        y--;
    }
    h = (d + 26 * (m + 1) / 10 + y + y / 4-y /100 + y /400 ) % 7 ;
    switch(h)
    {
    case 0:
        printf("Saturday\n");
```

```
        break;
    case 1:
        printf("Sunday\n");
        break;
    case 2:
        printf("Monday\n");
        break;
    case 3:
        printf("Tuesday\n");
        break;
    case 4:
        printf("Wednesday\n");
        break;
    case 5:
        printf("Thursday\n");
        break;
    case 6:
        printf("Friday\n");
    }
}
```

```
void AddNote()
{
    char *p;
    char date[50],event[300];
    FILE *fp = fopen("event.txt","a");
    printf("Enter date :\n");
    scanf("%s",&date);

    printf("Enter your Events :\n");
    scanf("%[^\t]",&event);
    if(p=strchr(event,'\n'!=NULL))
       *p='\n';
   fprintf(fp,"\n----------------------------\n");
    fprintf(fp,"Date :\n%s",date);
    fprintf(fp,"\nEvent :%s",event);
  fprintf(fp,"\n----------------------------\n");
    fclose(fp);
}
```

```
void ViewNote()
{
    FILE *r = fopen("event.txt","r");
   char c;
   while(1)
   {
      c = fgetc(r);
      if(c==EOF)
         break;
      printf("%c",c);
   }
   fclose(r);
}
```

```
void AddNoteBangla()
{
    char *p;
    char date[50],note[300];
    FILE *fp = fopen("note.txt","a");
    printf("Enter date :\n");
    scanf("%s",&date);

    printf("Enter your Events :\n");
    scanf("%[^\t]",&note);
    if(p=strchr(note,'\n'!=NULL))
       *p='\n';
   fprintf(fp,"\n----------------------------\n");
    fprintf(fp,"Date :\n%s",date);
    fprintf(fp,"\nEvent :%s",note);
  fprintf(fp,"\n----------------------------\n");
    fclose(fp);
}
```

```
void ViewNoteBangla()
{
    FILE *r = fopen("note.txt","r");
   char c;
   while(1)
   {
      c = fgetc(r);
      if(c==EOF)
         break;
      printf("%c",c);
   }
   fclose(r);
}
```

```
//Main Function
int main()
{
   system("Color 5F");
   int x;
menu:
   {
      printf("Which Calender do you want to see ?\n\n");
      printf("1.English Calender \n");
      printf("2.Bangla Calender \n");
      scanf("%d",&x);
   }
   system("cls");
   if(x==1) {
      system("Color E");
      int n,month;
      printf("Enter Your Choice        : \n\n");
      printf("1.Printing Specific Date  :\n");
      printf("2.Printing Specific Month :\n");
      printf("3.Printing Full Year \n");
```

```
printf("4.Add Note              :\n");
printf("5.View Note             :\n");

scanf("%d",&n);
system("cls");

if(n==1)
{
   system("Color 3F");
   int d,m,y;
   printf("Enter Specific Date DD/MM/YYYY : ");
   scanf("%d %d %d",&d,&m,&y);
   PrintingSpecificDate(d,m,y);
}

else if (n==2)
{
   system("Color 4E");
   printf("\n\nEnter which Year's Month you want to see :\n");
   scanf("%d",&year);
```

```
        printf("\nEnter Month that you want to see :\n");

printf("1.January\n2.February\n3.March\n4.April\n5.May\n6.June\n7.July\n8.August\n9.September\n10.October\n11.November\n12.December\n");
        scanf("%d",&month);
        PrintingMonth(month-1);
    }
    else if (n==3)
    {
        system("Color 8E");
        //Taking input a specific year
        printf("\n\nEnter a year that you want to see :");
        scanf("%d",&year);
        PrintingYear();
    }
    else if(n==4)
    {
        AddNote();
    }
    else if(n==5)
```

```
    {
        ViewNote();
    }
    else
        printf("Wrong Input");
}
else if(x==2)
{
    system("Color 1F");
    int n,month;
    printf("Enter Your Choice         : \n\n");
    printf("1.Printing Specific Date  :\n");
    printf("2.Printing Specific Month :\n");
    printf("3.Printing Full Year      :\n");
    printf("4.Add Note                :\n");
    printf("5.View Note               :\n");

    scanf("%d",&n);
    system("cls");
    if(n==1)
```

```
    {
        system("Color 3F");
        int d,m,y;
        printf("Enter DD/MM/YYYY : ");
        scanf("%d %d %d",&d,&m,&y);
        PrintingSpecificDate(d,m,y);
    }

    else if (n==2)
    {
        system("Color 4E");
        printf("\n\nEnter which Bangla Year's Month you want to see
:\n");
        scanf("%d",&year);
        printf("\nEnter Bangla Month that you want to see :\n");

printf("1.Baishakh\n2.Jaishtha\n3.Ashar\n4.Srabon\n5.Bhadra\n6.Ashwi
n\n7.Karttik\n8.Agrahayan\n9.Paush\n10.Magh\n11.Falgun\n12.Chaitra\
n");
        scanf("%d",&month);
        PrintingBanglaMonth(month-1);
    }
```

```
    else if (n==3)
    {
        system("Color 8E");
        //Taking input a specific year
        printf("\n\nEnter a Bangla year that you want to see :");
        scanf("%d",&year);
        PrintingBanglaYear();
    }
    else if(n==4)
    {
        AddNoteBangla();
    }
    else if(n==5)
    {
        ViewNoteBangla();
    }
    else
        printf("Wrong Input");
}
else
```

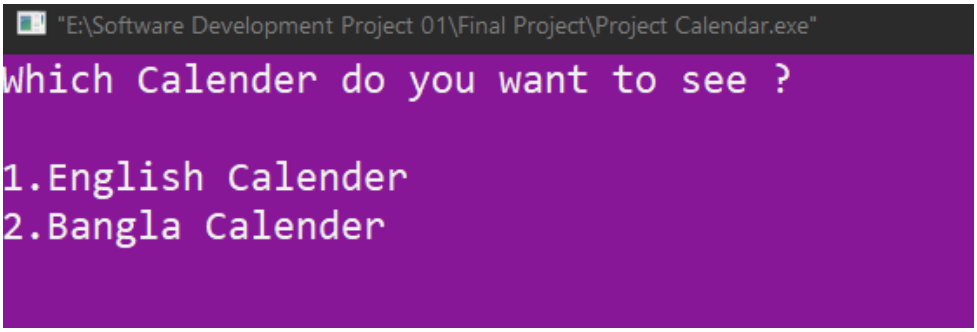
```
        printf("Wrong Input");
    int w;
    printf("\n\n\n Enter Your Choice : \n 1.Main Menu \n 2.Exit\n");
    scanf("%d",&w);
    system("cls");
    if(w==1)
    {
        goto menu;
    }
    else if (w=2)
    {
        return 0;
    }
    return 0;
}
```

Chapter – 6

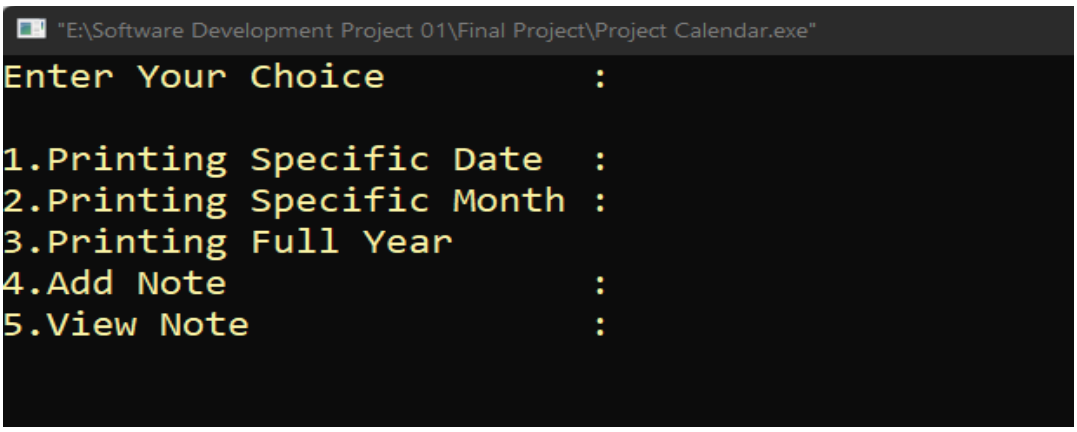
6.1 - Output

Output:

1.Printing type of Calendar:

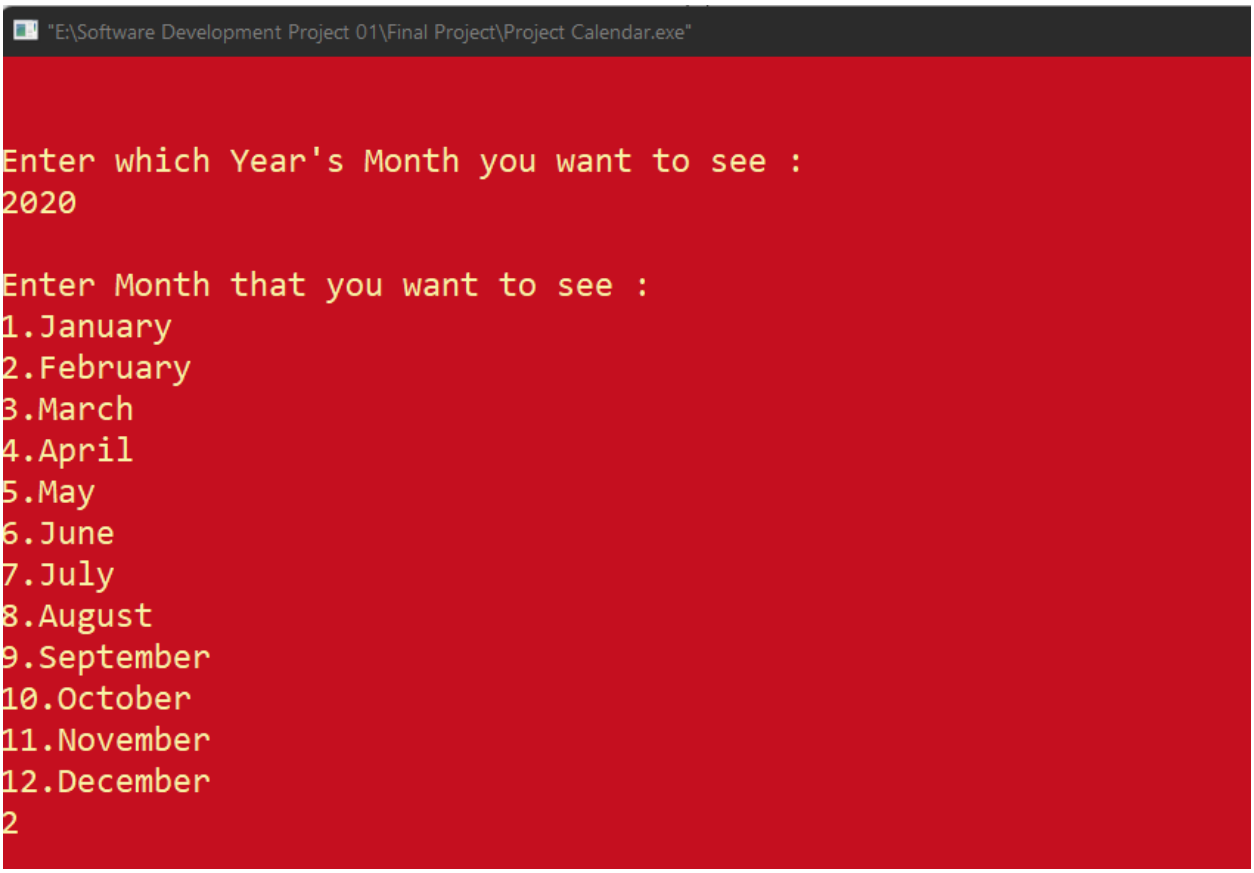


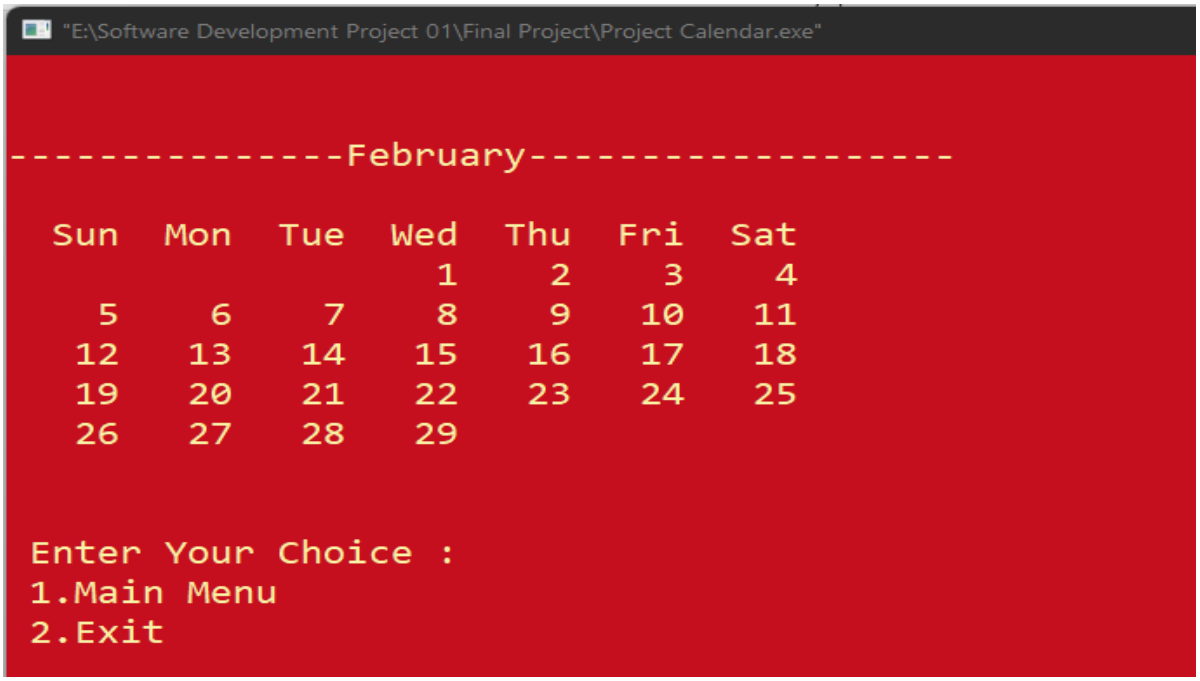
02.Choice menu of English Calendar :



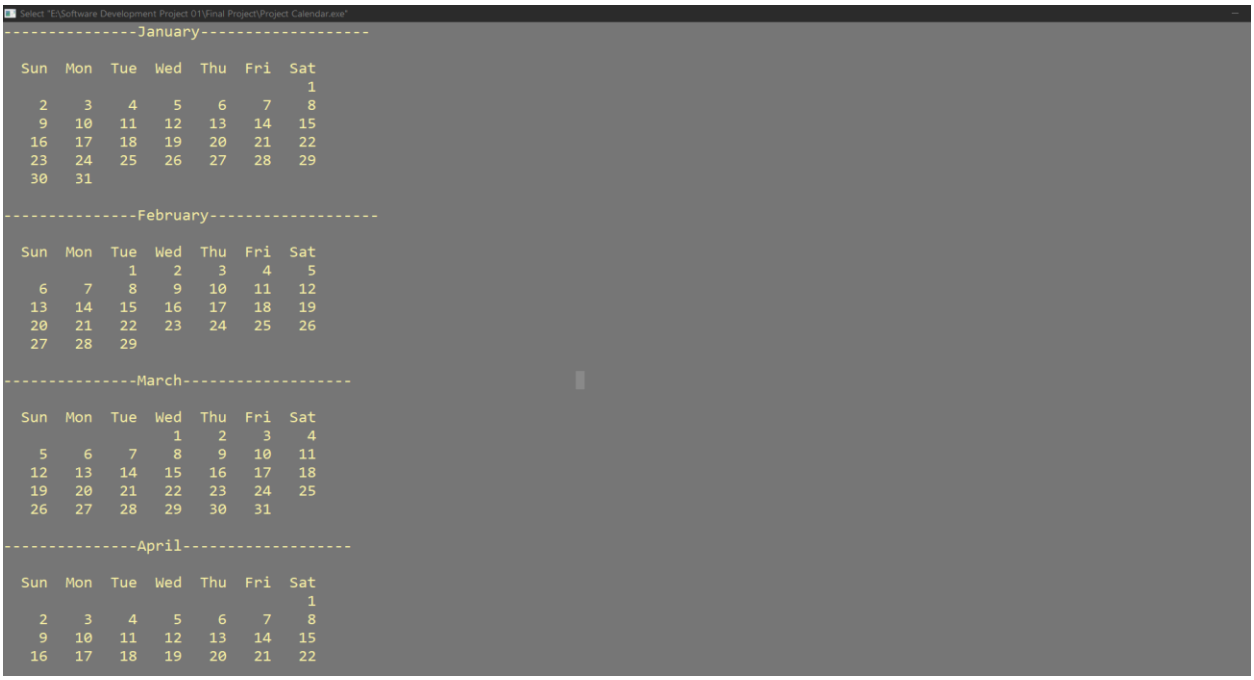
03.Printing Specific Date:

04.Printing Specific Month:



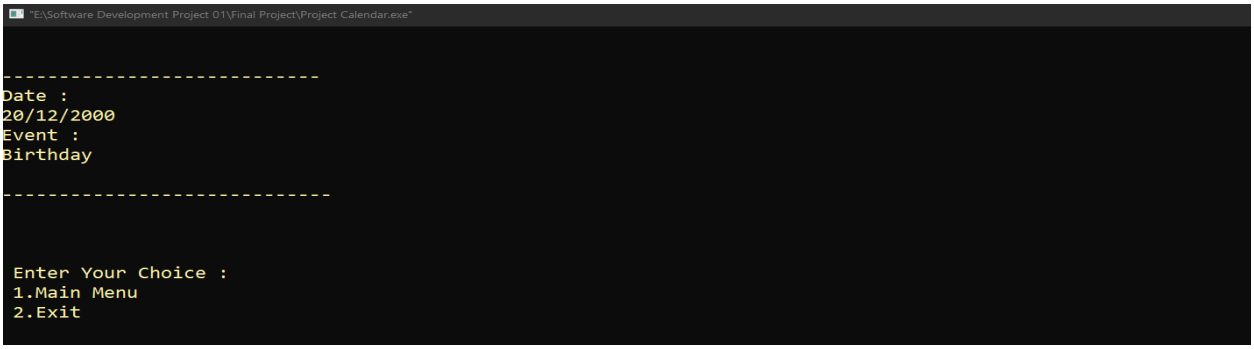


05.Printing Full Year:



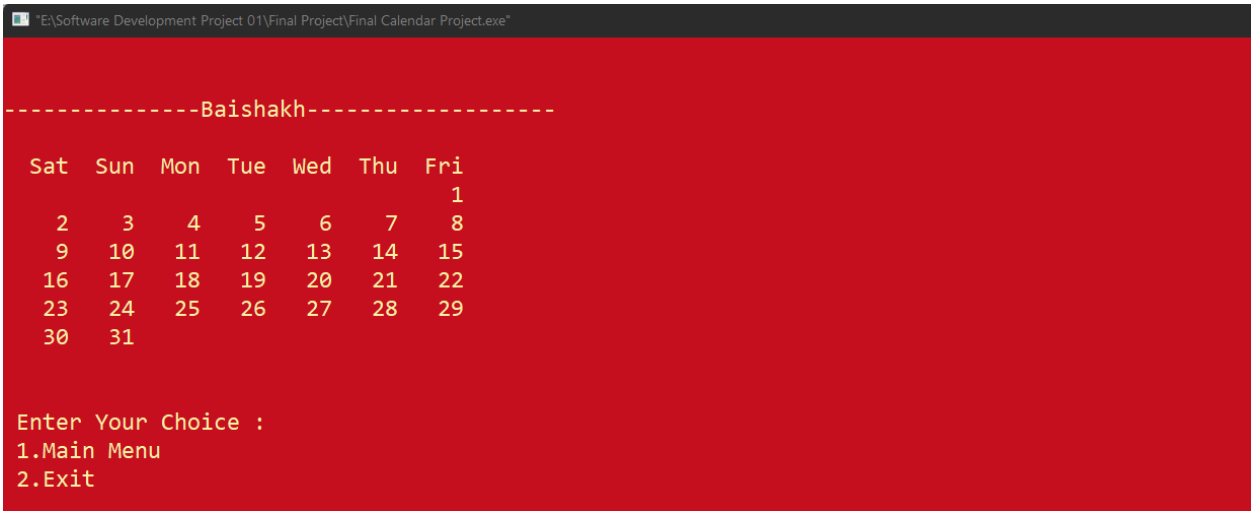
06.Adding Note:

07.View Note:

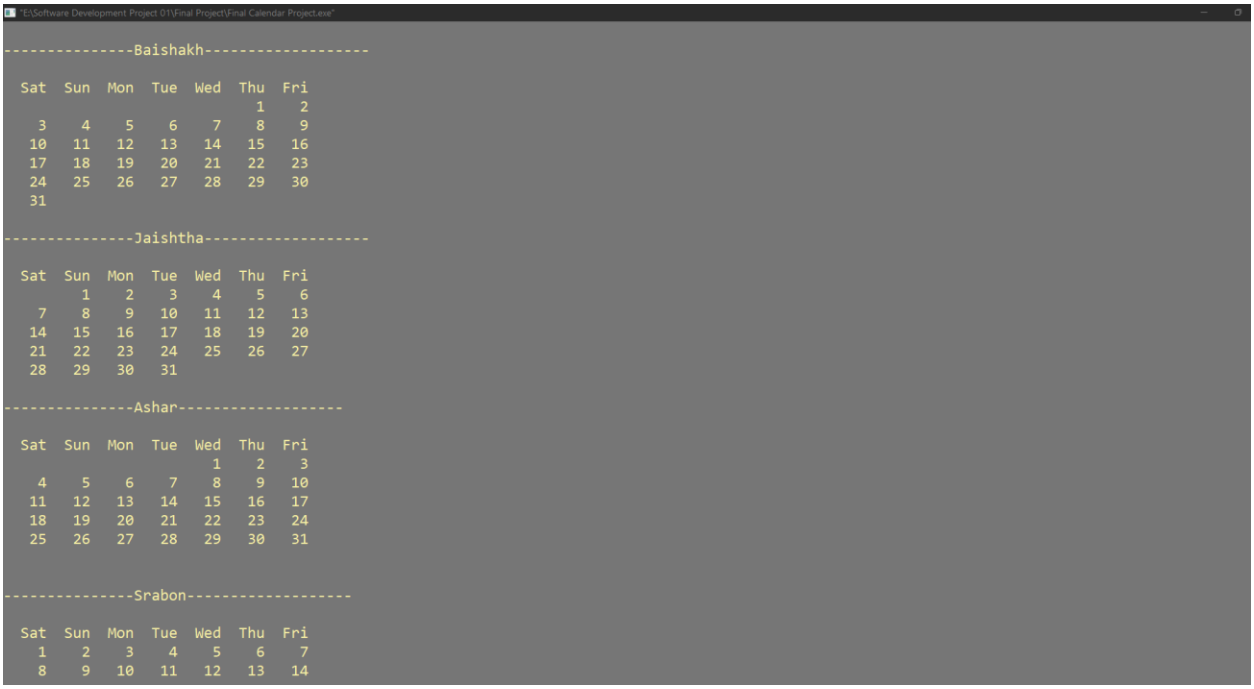


08.Printing Specific Bangla date:

09.Printing Specific Bangla month:

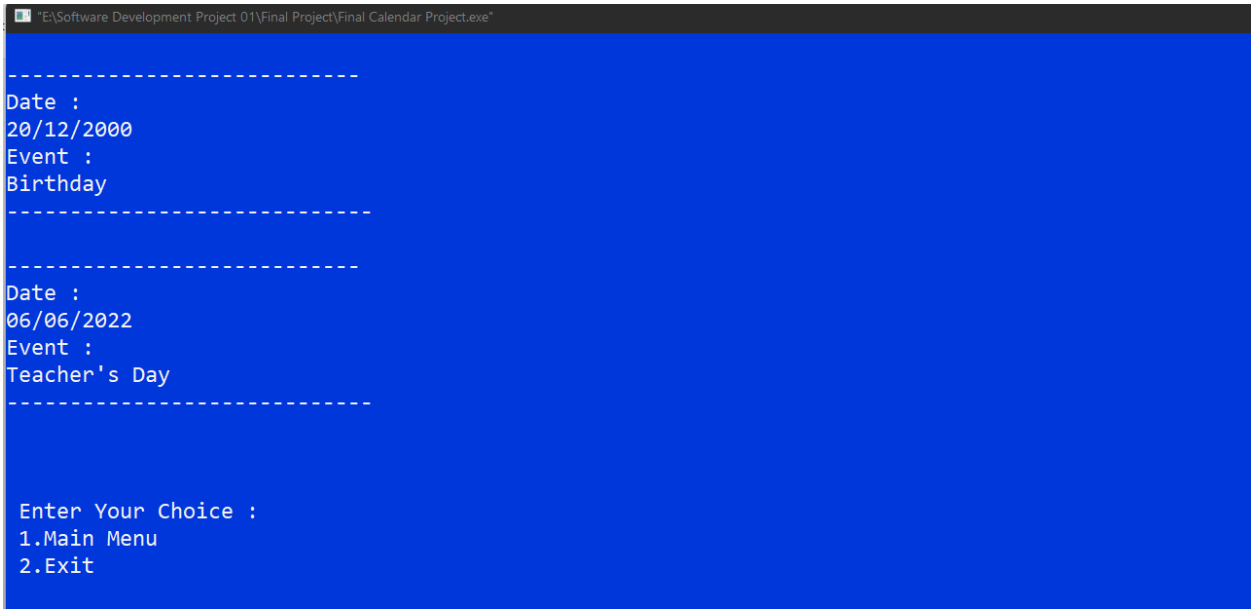


10.Printing full year:



11.Adding Note:

12.Viewing Note:



Chapter – 7

7.1 – Conclusion

For developing this project we faced some difficulties which are solved by the directions of our honorable supervisor sir. We are still working on it for adding some additional features to make this project more user friendly.

7.2 – Limitation

A lot of limitation has in our project. We have not set any alarm or reminder for any special events. So this project can't remind us any events. This calendar can't mark any holidays. This calendar can't display both Bangla and English calendar at a same time.

7.3 – Future work

We are thinking about to include alarm and reminder feature in this calendar. And we are also thinking about marking holidays. At last we want to include all types of calendars which are exist all over the world.

Reference

1.About Programming Language

Available Link: https://www.javatpoint.com/programming-language

2.About C Programming

Available Link: https://www.geeksforgeeks.org/c-language-set-1-introduction/

3.Calendar Idea

Available Link: https://www.youtube.com/watch?v=VRcJpUOKRt8&ab_channel=TechsaneGirl