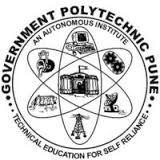
Government Polytechnic, Pune-16

**(An Autonomous Institute of Government of Maharashtra)**



**A**

**Micro project Report**

**On**

# “Calender application”

**SUBMITTED BY:**

|  |  |
| --- | --- |
| **SAEID MULANI** | **2007041** |
| **SANIYA PATHAN** | **2007048** |
| **SIDDHESH PATIL** | **2007051** |

**Under the Guidance of**

**“SHRADDHA HANDE MAM”**

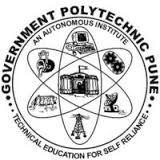
**DEPARTMENT OF INFORMATION TECHNOLOGY**

**(Academic Year: 2021-22)**

## Government Polytechnic, Pune-16

**(An Autonomous Institute of Government of Maharashtra)**

### **Department Information Technology**



## CERTIFICATE

**This is to certify that Saeid Mulani(2007041), Saniya Pathan (2007048) & Siddhesh Patil (2007051) of Second Year Diploma in Information Technology has successfully completed the Micro project titled “ CALENDER APPLICATION” as part of his diploma curriculum of Subject :- DATA STRUCTURE in academic year 2021-22.**

**Guide HOD Principal**

**( Smt.Shraddha Hande) (Smt. M.U Kokate) (Dr. V.S Bandal)**

**ACKNOWLEDGMENT**

It is my proud privilege and duty to acknowledge the kind of help and guidance received from several people in the preparation of this report. It would not have been possible to prepare this report in this form without their valuable help, cooperation and guidance.

First and foremost I express my sincere gratitude to my Guide, **Smt.Shraddha Hande** for guiding me in investigations of this seminar and in carrying out experimental work. My sincere thanks to **Mrs .M. U. Kokate.** Head of the department, Information Technology, Government Polytechnic Pune for her valuable suggestions and guidance throughout the preparation of this report.

Last but not the least I wish to thank my parents for financing my studies and helping me throughout my life for achieving perfection and excellence. Their personal help in making this report and seminar worth presentation is gratefully acknowledge. **- SAEID MULANI (2007041), SANIYA PATHAN (2007048) & SIDDHESH PATIL(2007051).**

**PROBLEM STATEMENT**

The general problems to be addressed by the Calendar Tool project are the following:

1. to provide a reliable and easy to use that manages an individual user's calendar
2. to provide a secure group environment that manages a collection of individual calendars for the purposes of group scheduling and information sharing.

While these problems have been solved by a number of commercially available and public domain products, the **“CALENDER APPLICATION**” provides a solution with certain functional improvements over existing tools. Specific problem areas are the following:

1. scheduling multi-user meetings
2. dealing effectively with overlapping events
3. providing a wide range of calendar viewing options

The Calendar application project addresses the requirements of a particular people.

**ABSTRACT**

Calendar applications which accesses the calendars of multiple users, for instance members of a family or staff in a particular department and will pull out the events from every calendar. Hence, everyone in the group can see others’ events on one screen in customized format. It also helps the user to create, edit, and delete events of his/her calendar. Additionally, this application will use an efficient strategy to update data only if there is any change in the events, making it faster. There are many other applications in the Play store but they cannot get rid of redundant events and reminders of events belonging to other users. This causes a lot of disturbance to the user. On the contrary, Calendar+ application will use an intelligent algorithm to compare the events in many ways based on the time, name, and place of event. Furthermore, this application will only generate reminders for the user events.

**SOURCE CODE**

------------------------------------------------------------------------------------------------------------

#include<stdio.h>

#include<conio.h>

#include<windows.h>

struct Date{

int dd;

int mm;

int yy;

};

struct Date date;

struct Remainder{

int dd;

int mm;

char note[50];

};

struct Remainder R;

COORD xy = {0, 0};

void gotoxy (int x, int y)

{

xy.X = x; xy.Y = y; // X and Y coordinates

SetConsoleCursorPosition(GetStdHandle(STD\_OUTPUT\_HANDLE), xy);

}

//This will set the forground color for printing in a console window.

void SetColor(int ForgC)

{

WORD wColor;

//We will need this handle to get the current background attribute

HANDLE hStdOut = GetStdHandle(STD\_OUTPUT\_HANDLE);

CONSOLE\_SCREEN\_BUFFER\_INFO csbi;

//We use csbi for the wAttributes word.

if(GetConsoleScreenBufferInfo(hStdOut, &csbi))

{

//Mask out all but the background attribute, and add in the forgournd color

wColor = (csbi.wAttributes & 0xF0) + (ForgC & 0x0F);

SetConsoleTextAttribute(hStdOut, wColor);

}

return;

}

void ClearColor(){

SetColor(15);

}

void ClearConsoleToColors(int ForgC, int BackC)

{

WORD wColor = ((BackC & 0x0F) << 4) + (ForgC & 0x0F);

//Get the handle to the current output buffer...

HANDLE hStdOut = GetStdHandle(STD\_OUTPUT\_HANDLE);

//This is used to reset the carat/cursor to the top left.

COORD coord = {0, 0};

//A return value... indicating how many chars were written

// not used but we need to capture this since it will be

// written anyway (passing NULL causes an access violation).

DWORD count;

//This is a structure containing all of the console info

// it is used here to find the size of the console.

CONSOLE\_SCREEN\_BUFFER\_INFO csbi;

//Here we will set the current color

SetConsoleTextAttribute(hStdOut, wColor);

if(GetConsoleScreenBufferInfo(hStdOut, &csbi))

{

//This fills the buffer with a given character (in this case 32=space).

FillConsoleOutputCharacter(hStdOut, (TCHAR) 32, csbi.dwSize.X \* csbi.dwSize.Y, coord, &count);

FillConsoleOutputAttribute(hStdOut, csbi.wAttributes, csbi.dwSize.X \* csbi.dwSize.Y, coord, &count );

//This will set our cursor position for the next print statement.

SetConsoleCursorPosition(hStdOut, coord);

}

return;

}

void SetColorAndBackground(int ForgC, int BackC)

{

WORD wColor = ((BackC & 0x0F) << 4) + (ForgC & 0x0F);;

SetConsoleTextAttribute(GetStdHandle(STD\_OUTPUT\_HANDLE), wColor);

return;

}

int check\_leapYear(int year){ //checks whether the year passed is leap year or not

if(year % 400 == 0 || (year % 100!=0 && year % 4 ==0))

return 1;

return 0;

}

void increase\_month(int \*mm, int \*yy){ //increase the month by one

++\*mm;

if(\*mm > 12){

++\*yy;

\*mm = \*mm - 12;

}

}

void decrease\_month(int \*mm, int \*yy){ //decrease the month by one

--\*mm;

if(\*mm < 1){

--\*yy;

if(\*yy<1600){

printf("No record available");

return;

}

\*mm = \*mm + 12;

}

}

int getNumberOfDays(int month,int year){ //returns the number of days in given month

switch(month){ //and year

case 1 : return(31);

case 2 : if(check\_leapYear(year)==1)

return(29);

else

return(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(-1);

}

}

char \*getName(int day){ //returns the name of the day

switch(day){

case 0 :return("Sunday");

case 1 :return("Monday");

case 2 :return("Tuesday");

case 3 :return("Wednesday");

case 4 :return("Thursday");

case 5 :return("Friday");

case 6 :return("Saturday");

default:return("Error in getName() module.Invalid argument passed");

}

}

void print\_date(int mm, int yy){ //prints the name of month and year

printf("---------------------------\n");

gotoxy(25,6);

switch(mm){

case 1: printf("January"); break;

case 2: printf("February"); break;

case 3: printf("March"); break;

case 4: printf("April"); break;

case 5: printf("May"); break;

case 6: printf("June"); break;

case 7: printf("July"); break;

case 8: printf("August"); break;

case 9: printf("September"); break;

case 10: printf("October"); break;

case 11: printf("November"); break;

case 12: printf("December"); break;

}

printf(" , %d", yy);

gotoxy(20,7);

printf("---------------------------");

}

int getDayNumber(int day,int mon,int year){ //retuns the day number

int res = 0, t1, t2, y = year;

year = year - 1600;

while(year >= 100){

res = res + 5;

year = year - 100;

}

res = (res % 7);

t1 = ((year - 1) / 4);

t2 = (year-1)-t1;

t1 = (t1\*2)+t2;

t1 = (t1%7);

res = res + t1;

res = res%7;

t2 = 0;

for(t1 = 1;t1 < mon; t1++){

t2 += getNumberOfDays(t1,y);

}

t2 = t2 + day;

t2 = t2 % 7;

res = res + t2;

res = res % 7;

if(y > 2000)

res = res + 1;

res = res % 7;

return res;

}

char \*getDay(int dd,int mm,int yy){

int day;

if(!(mm>=1 && mm<=12)){

return("Invalid month value");

}

if(!(dd>=1 && dd<=getNumberOfDays(mm,yy))){

return("Invalid date");

}

if(yy>=1600){

day = getDayNumber(dd,mm,yy);

day = day%7;

return(getName(day));

}else{

return("Please give year more than 1600");

}

}

int checkNote(int dd, int mm){

FILE \*fp;

fp = fopen("note.dat","rb");

if(fp == NULL){

printf("Error in Opening the file");

}

while(fread(&R,sizeof(R),1,fp) == 1){

if(R.dd == dd && R.mm == mm){

fclose(fp);

return 1;

}

}

fclose(fp);

return 0;

}

void printMonth(int mon,int year,int x,int y){ //prints the month with all days

int nod, day, cnt, d = 1, x1 = x, y1 = y, isNote = 0;

if(!(mon>=1 && mon<=12)){

printf("INVALID MONTH");

getch();

return;

}

if(!(year>=1600)){

printf("INVALID YEAR");

getch();

return;

}

gotoxy(20,y);

print\_date(mon,year);

y += 3;

gotoxy(x,y);

printf("S M T W T F S ");

y++;

nod = getNumberOfDays(mon,year);

day = getDayNumber(d,mon,year);

switch(day){ //locates the starting day in calender

case 0 :

x=x;

cnt=1;

break;

case 1 :

x=x+4;

cnt=2;

break;

case 2 :

x=x+8;

cnt=3;

break;

case 3 :

x=x+12;

cnt=4;

break;

case 4 :

x=x+16;

cnt=5;

break;

case 5 :

x=x+20;

cnt=6;

break;

case 6 :

x=x+24;

cnt=7;

break;

default :

printf("INVALID DATA FROM THE getOddNumber()MODULE");

return;

}

gotoxy(x,y);

if(cnt == 1){

SetColor(12);

}

if(checkNote(d,mon)==1){

SetColorAndBackground(15,12);

}

printf("%02d",d);

SetColorAndBackground(15,1);

for(d=2;d<=nod;d++){

if(cnt%7==0){

y++;

cnt=0;

x=x1-4;

}

x = x+4;

cnt++;

gotoxy(x,y);

if(cnt==1){

SetColor(12);

}else{

ClearColor();

}

if(checkNote(d,mon)==1){

SetColorAndBackground(15,12);

}

printf("%02d",d);

SetColorAndBackground(15,1);

}

gotoxy(8, y+2);

SetColor(14);

printf("Press 'n' to Next, Press 'p' to Previous and 'q' to Quit");

gotoxy(8,y+3);

printf("Red Background indicates the NOTE, Press 's' to see note: ");

ClearColor();

}

void AddNote(){

FILE \*fp;

fp = fopen("note.dat","ab+");

system("cls");

gotoxy(5,7);

printf("Enter the date(DD/MM): ");

scanf("%d%d",&R.dd, &R.mm);

gotoxy(5,8);

printf("Enter the Note(50 character max): ");

fflush(stdin);

scanf("%[^\n]",R.note);

if(fwrite(&R,sizeof(R),1,fp)){

gotoxy(5,12);

puts("Note is saved sucessfully");

fclose(fp);

}else{

gotoxy(5,12);

SetColor(12);

puts("\aFail to save!!\a");

ClearColor();

}

gotoxy(5,15);

printf("Press any key............");

getch();

fclose(fp);

}

void showNote(int mm){

FILE \*fp;

int i = 0, isFound = 0;

system("cls");

fp = fopen("note.dat","rb");

if(fp == NULL){

printf("Error in opening the file");

}

while(fread(&R,sizeof(R),1,fp) == 1){

if(R.mm == mm){

gotoxy(10,5+i);

printf("Note %d Day = %d: %s", i+1, R.dd, R.note);

isFound = 1;

i++;

}

}

if(isFound == 0){

gotoxy(10,5);

printf("This Month contains no note");

}

gotoxy(10,7+i);

printf("Press any key to back.......");

getch();

}

int main(){

ClearConsoleToColors(15, 1);

SetConsoleTitle("Calender Project - Programming-technique.blogspot.com");

int choice;

char ch = 'a';

while(1){

system("cls");

printf("1. Find Out the Day\n");

printf("2. Print all the day of month\n");

printf("3. Add Note\n");

printf("4. EXIT\n");

printf("ENTER YOUR CHOICE : ");

scanf("%d",&choice);

system("cls");

switch(choice){

case 1:

printf("Enter date (DD MM YYYY) : ");

scanf("%d %d %d",&date.dd,&date.mm,&date.yy);

printf("Day is : %s",getDay(date.dd,date.mm,date.yy));

printf("\nPress any key to continue......");

getch();

break;

case 2 :

printf("Enter month and year (MM YYYY) : ");

scanf("%d %d",&date.mm,&date.yy);

system("cls");

while(ch!='q'){

printMonth(date.mm,date.yy,20,5);

ch = getch();

if(ch == 'n'){

increase\_month(&date.mm,&date.yy);

system("cls");

printMonth(date.mm,date.yy,20,5);

}else if(ch == 'p'){

decrease\_month(&date.mm,&date.yy);

system("cls");

printMonth(date.mm,date.yy,20,5);

}else if(ch == 's'){

showNote(date.mm);

system("cls");

}

}

break;

case 3:

AddNote();

break;

case 4 :

exit(0);

}

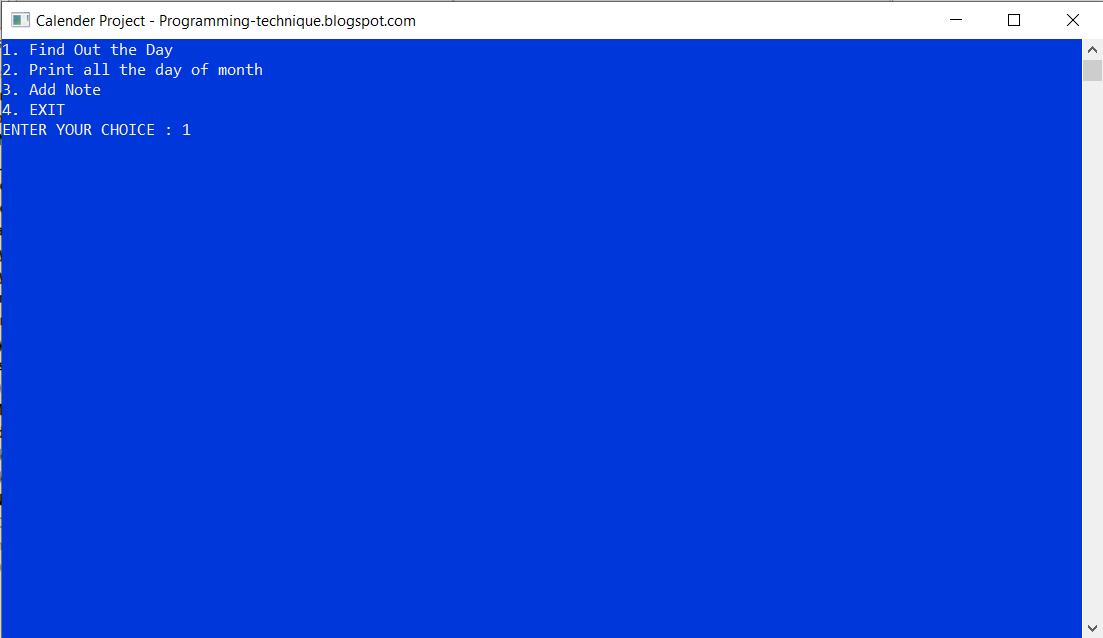
}

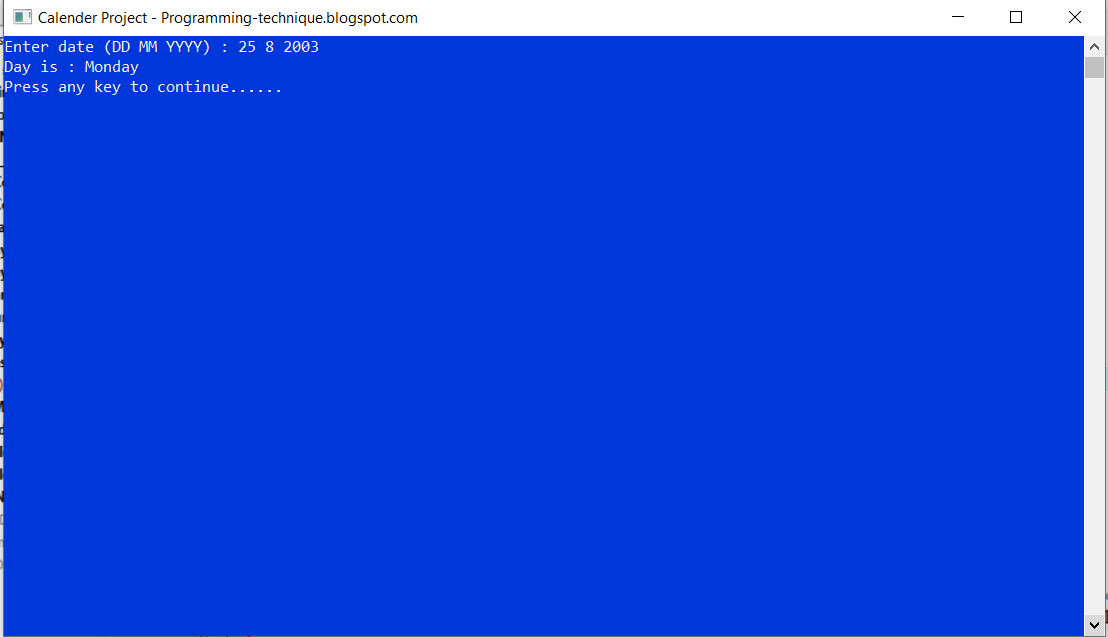
return 0;

**OUTPUT**

----------------------------------------------------------------------------------------------------------------

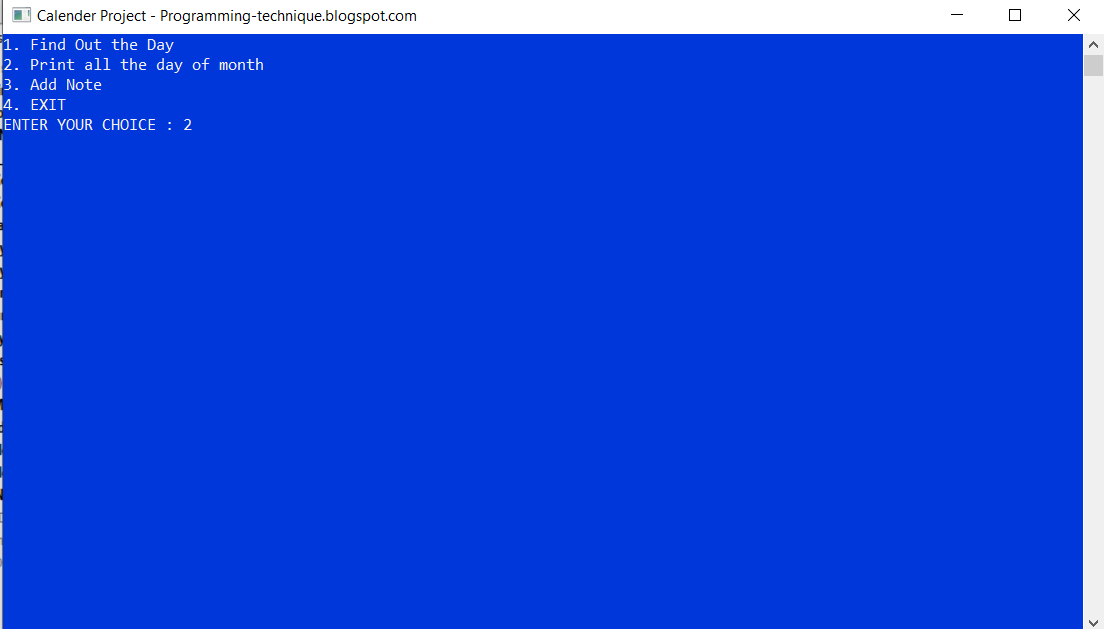
**1.FIND OUT THE DAY**

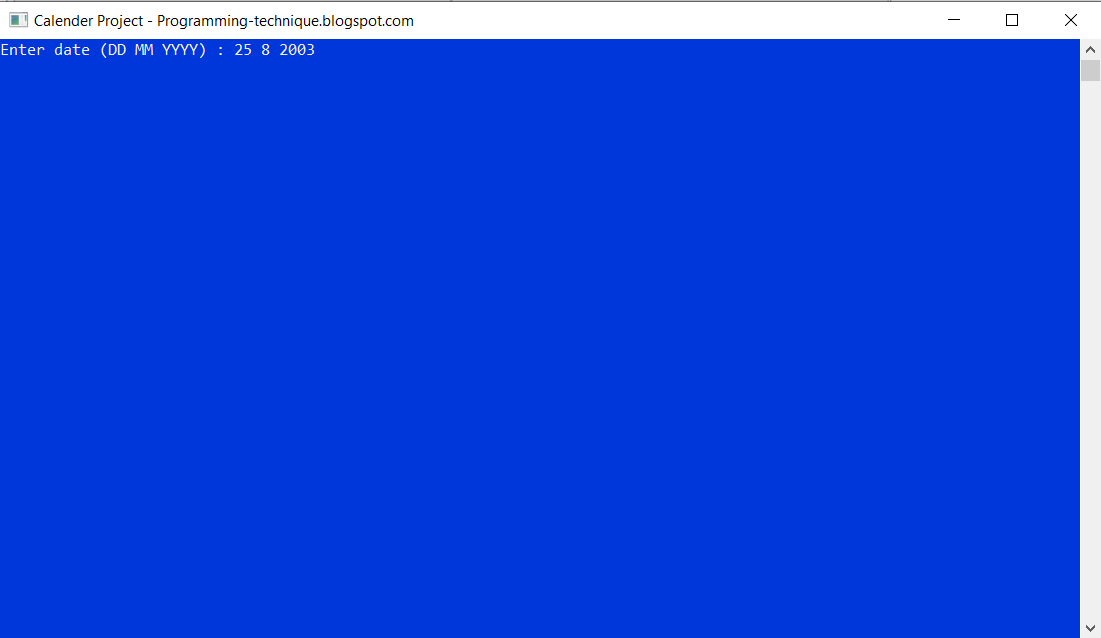


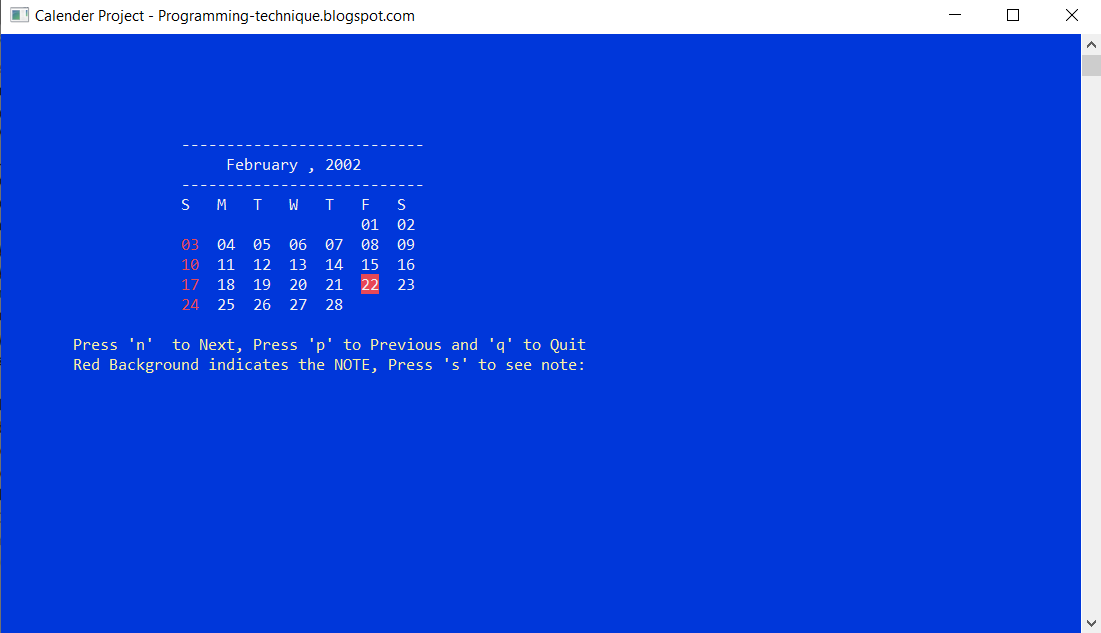
****

**2. PRINT ALL THE DAY OF MONTH**

----------------------------------------------------------------------------------------------------------------

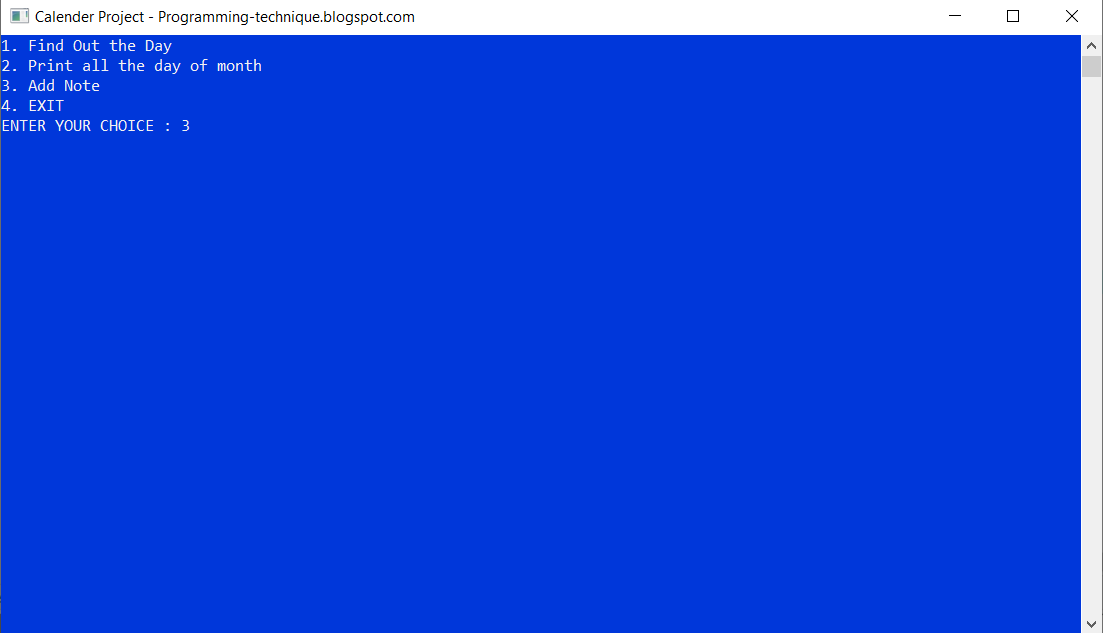
****

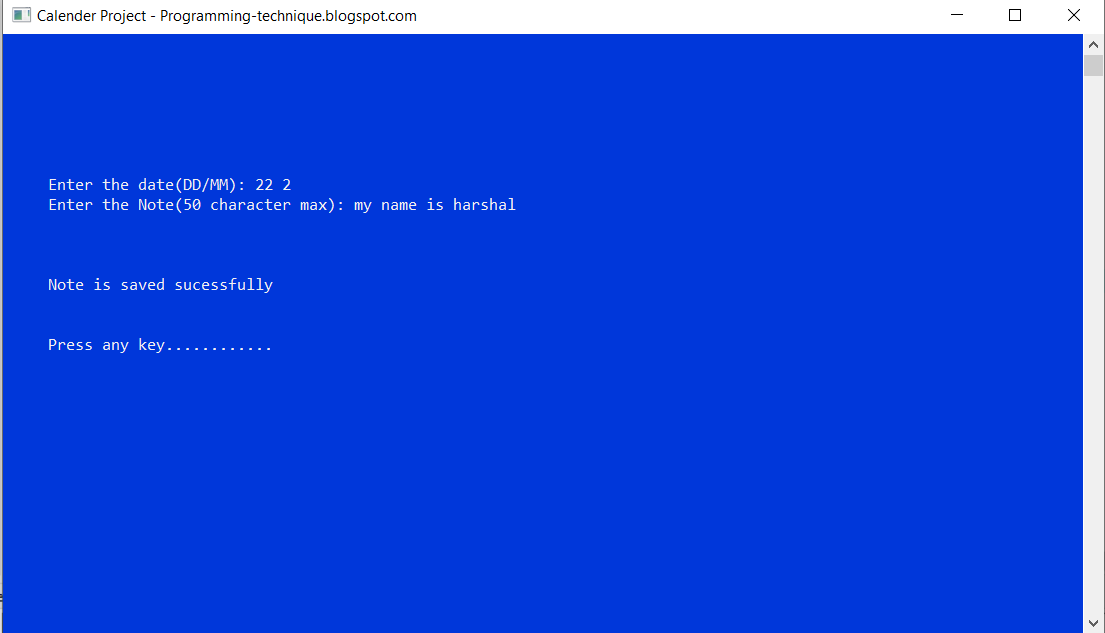
****

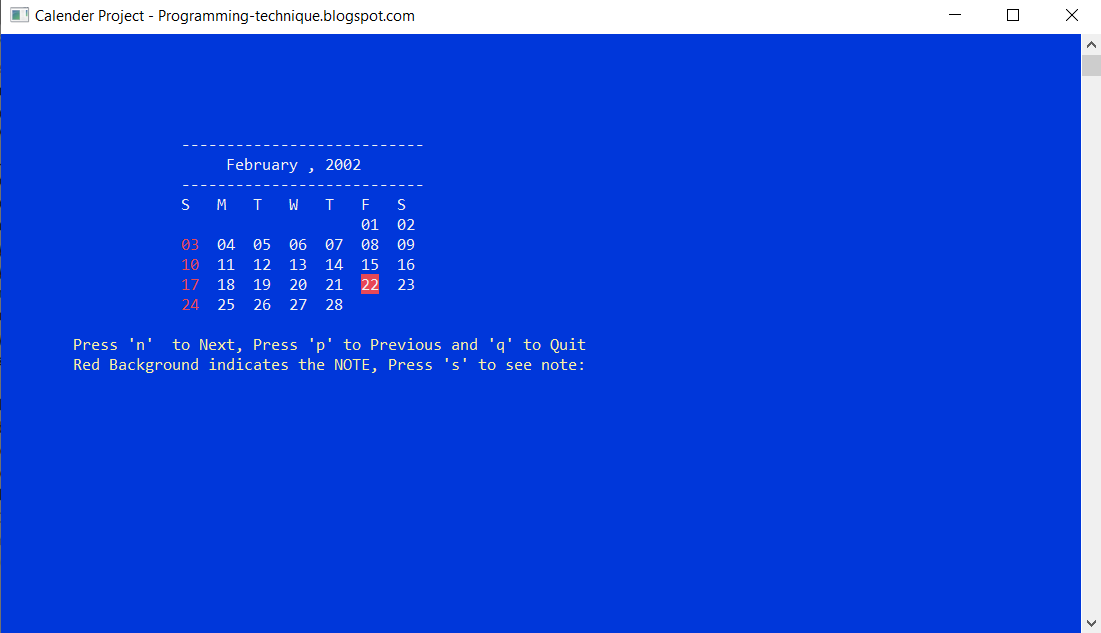
****

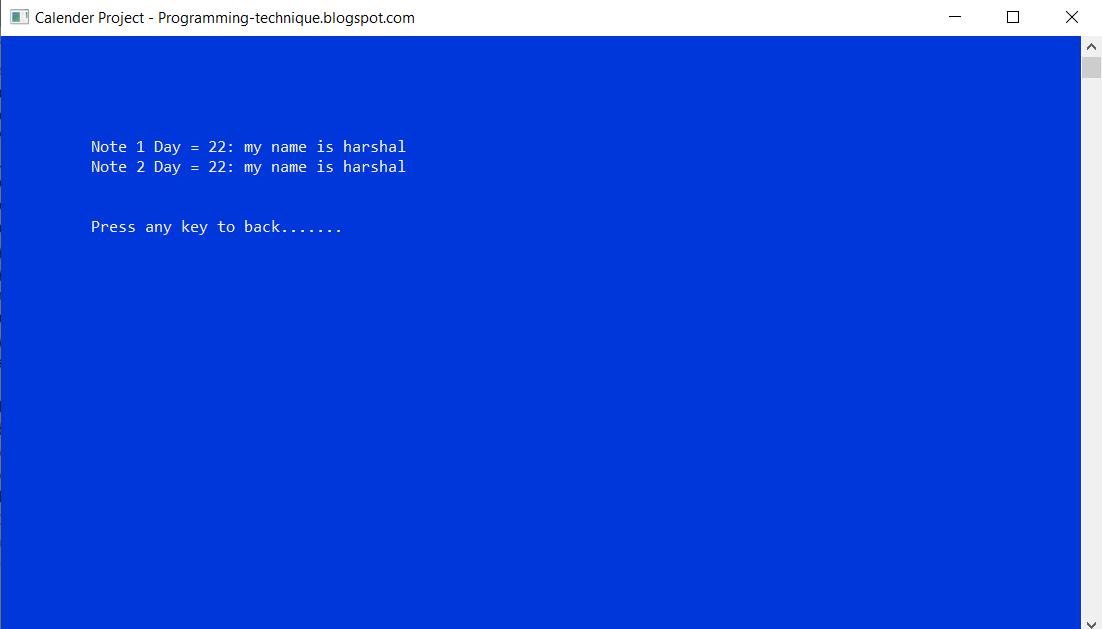
**3. ADD NOTE**

----------------------------------------------------------------------------------------------------------------

****

****

****

****

**CONCLUSION**

In this way, we have completed our project based on **“CALENDER APPLICATION”.** We are very thankful to Mrs. Hande ma’am for her guidance and support.

We tried our best to prepare this project informative and attractive as per given problem statement. The successful completion of our project would not have been possible without the dedicated support from all our mentors, family and friends. I am feeling satisfied after the completion of my project.