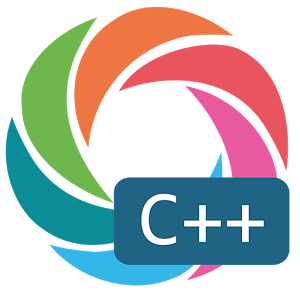
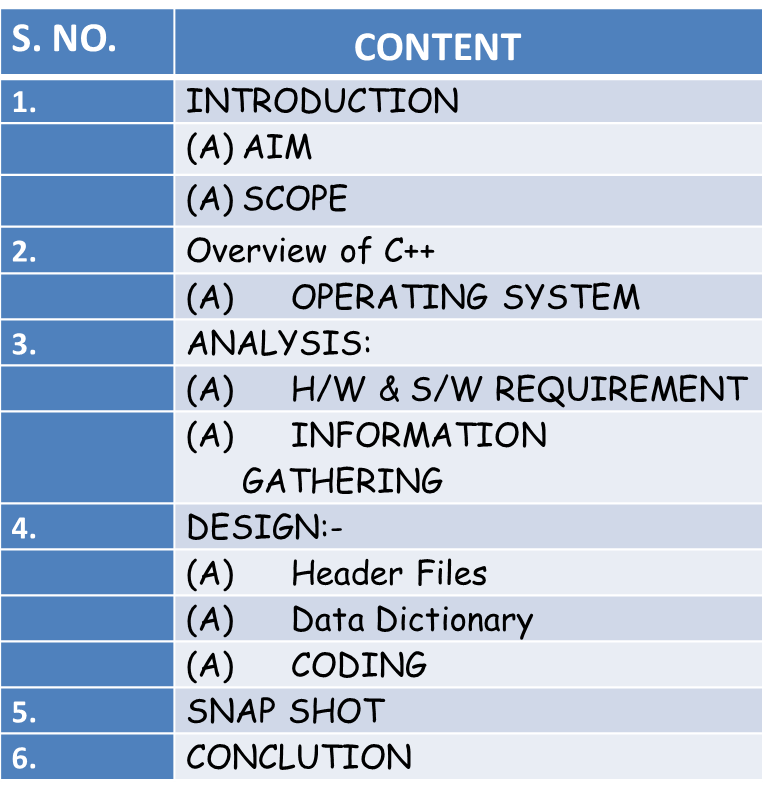
**CALCULATOR AND MATRIX MANIPULATOR**

**PROJECT BY AASHITA GLORIA NOAH (2017)**

**[](https://www.google.co.in/url?sa=i&rct=j&q=&esrc=s&source=images&cd=&cad=rja&uact=8&ved=0ahUKEwj7u7PbuNLYAhUJtI8KHWz4AlkQjRwIBw&url=https://play.google.com/store/apps/details?id=com.sololearn.cplusplus&hl=en&psig=AOvVaw2w32twI8mqoWmKiE21WoN_&ust=1515847113433683)**

**INDEX**

****

**AIM OF THE PROJECT**

The main aim of the project is to develop a program which performs the operations of a simple Calculator and builds a Matrix Manipulator which allows the user to input the matrix and perform operations such as addition, subtraction, multiplication, transpose and diagonal operations to find the sum of upper and lower diagonals and upper and lower triangular matrices

The different objectives of the project are:

1. To have a systematic and organized way of performing mathematical operations.

2. To minimize the manual work load.

3. To minimize human error

**SCOPE OF THE PROJECT**

The operations in a calculator and matrix manipulator can easily be performed by the user and it has various practical applications in real life situations.

* A calculator is used in homes, offices and institutions to simplify mathematical calculations
* A matrix is a rectangular array of numbers, symbols, or expressions, arranged in rows and columns which are used in physics related applications, matrices are applied in the study of electrical circuits, quantum mechanics and optics.

**OPERATING SYSTEMS**

This project may be used is platform independent which means that it works on any operating system like:

* Windows
* Linux
* Mac

**HARDWARE AND SOFTWARE REQUIREMENT**

**Minimum Hardware Requirement**

**Processor: PentiumIV or better.**

**Ram: 128 MB or more.**

**Hard Disk: 20 GB.**

**Monitor: Any.**

**Keyboard: 122 keys.**

**Minimum Software Requirement**

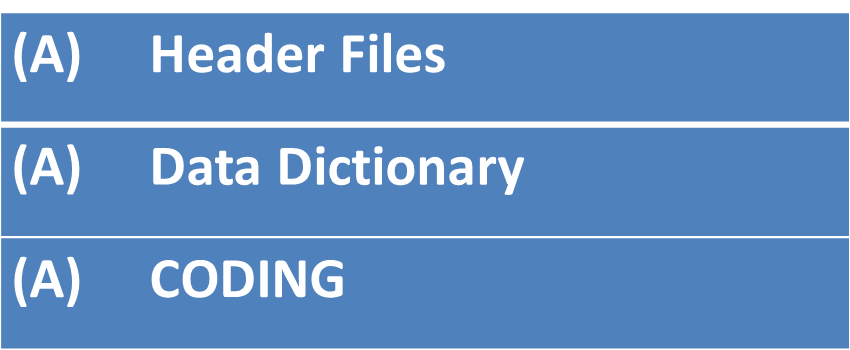
**Operating System: Windows 98, Windows XP, Windows 7 or better.**

**Language: C++.**

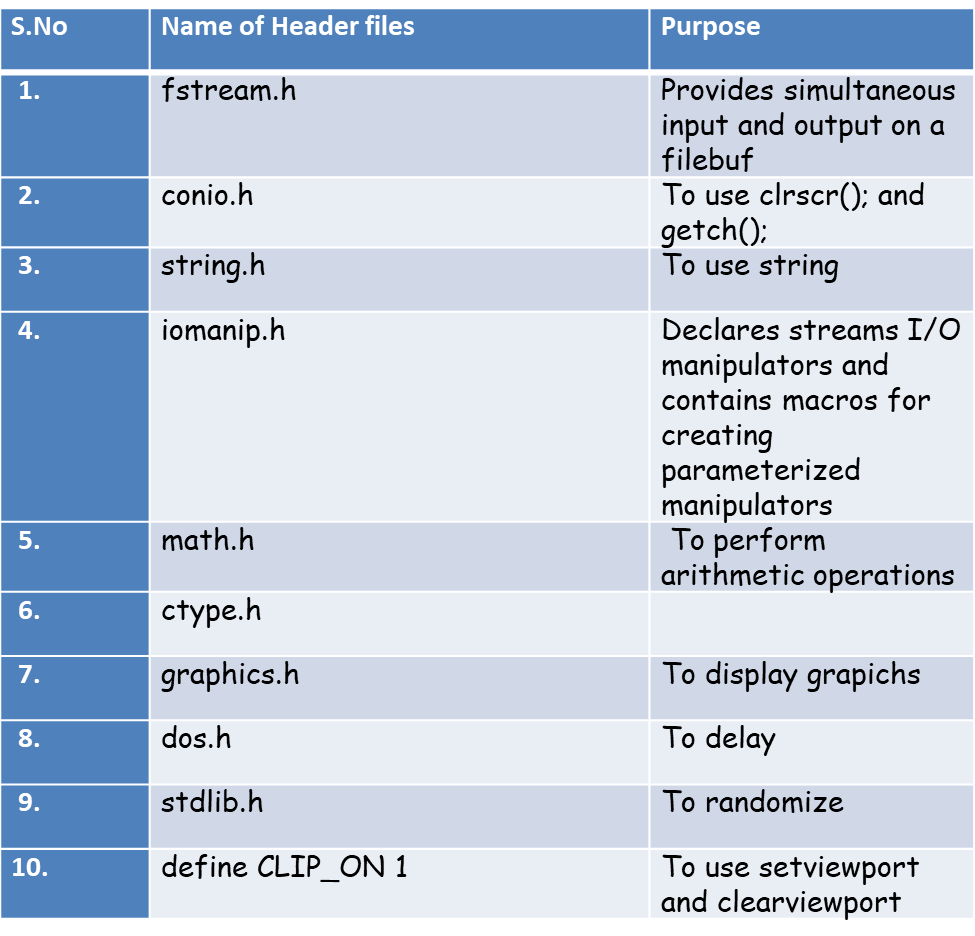
**Information Gathering :**

**Internet and text books**

**DESIGN**

****

1. **HEADER FILES**



1. **DATA DICTIONARY**

|  |  |  |
| --- | --- | --- |
| **S.No.** | **Name Of Functions** | **Purpose** |
| **1.** | float add  (int a,int b) | To perform addition of 2 numbers |
| **2.** | float substract | To perform subtraction of 2 numbers |
| **3.** | float multiply | To perform multiplication of 2 numbers |
| **4.** | float divide | To perform division of 2 numbers |
| **5.** | float scroot | To find square root of a number |
| **6.** | float power | To find the power of a number |
| **7.** | int modulus | To find the remainder by diving 2 numbers |
| **8.** | void calculate | To perform all arithmetic operations |
| **9.** | void calculator | To display the calculator screen |
| **10.** | void operation | To perform operation on matrices |
| **11.** | void matrix | To display input and display the matrix entered by the user |
| **12.** | void front | To display front page |
| **13.** | void iremark | To input remarks |
| **14.** | void display | To display name, age and remarks |

**SOURCE CODE:**

#include<iostream.h>

#include<fstream.h>

#include<conio.h>

#include<string.h>

#include<iomanip.h>

#include<math.h>

#include<ctype.h>

#include<graphics.h>

#include<dos.h>

#include<stdlib.h>

#include<stdio.h>

#define CLIP\_ON 1

//calculator

float add(float a,float b)

{

return (a+b);

}

float substract(float c,float d)

{

return (c-d);

}

float multiply(float e,float f)

{

return (e\*f);

}

float divide(float g,float h)

{

return (g/h);

}

float sroot(float i)

{

return(sqrt(i));

}

float power(float j,float k)

{

return(pow(j,k));

}

int modulus(int l,int m)

{

return(l%m);

}

void calculate(float x,char choice,float y=0)

{

float z;

if (choice == '+')

{

z=add(x,y);

clearviewport();

gotoxy(25,7);

cout<<z;

}

else if (choice == '-')

{

z=substract(x,y);

clearviewport();

gotoxy(25,7);

cout<<z;

}

else if (choice =='\*')

{

z=multiply(x,y);

clearviewport();

gotoxy(25,7);

cout<<z;

}

else if (choice =='/')

{

z=divide(x,y);

clearviewport();

gotoxy(25,7);

cout<<z;

}

else if (choice =='s')

{

z=sroot(x);

clearviewport();

gotoxy(25,7);

cout<<z;

}

else if (choice =='p')

{

z=power(x,y);

clearviewport();

gotoxy(25,7);

cout<<z;

}

else if (choice =='m')

{

z=modulus(x,y);

clearviewport();

gotoxy(25,7);

cout<<z;

}

}

void calculator()

{

clrscr();

int gd=DETECT,gm;

initgraph(&gd,&gm,"C:\\TURBOC3\\BGI");

setbkcolor(BLACK);

rectangle(120,80,450,400); //inside main

rectangle(110,70,460,410); //outside main

setfillstyle(11,DARKGRAY);

rectangle(130,90,440,140); //long one

rectangle(130,160,180,210);

rectangle(185,160,235,210);

rectangle(240,160,290,210);

rectangle(130,215,180,265);

rectangle(185,215,235,265);

rectangle(240,215,290,265);

rectangle(130,270,180,320);

rectangle(185,270,235,320);

rectangle(240,270,290,320);

rectangle(130,325,180,375);

rectangle(185,325,235,375);

rectangle(240,325,290,375);

rectangle(320,160,370,210);

rectangle(375,160,425,210);

rectangle(320,215,370,265);

rectangle(375,215,425,265);

rectangle(320,270,370,320);

rectangle(375,270,425,320);

rectangle(320,325,370,375);

rectangle(375,325,425,375);

settextstyle(3,0,3);

setcolor(MAGENTA);

outtextxy(140,170," 1 ");

outtextxy(195,170," 2 ");

outtextxy(250,170," 3 ");

outtextxy(140,225," 4 ");

outtextxy(195,225," 5 ");

outtextxy(250,225," 6 ");

outtextxy(140,280," 7 ");

outtextxy(195,280," 8 ");

outtextxy(250,280," 9 ");

outtextxy(195,335," 0 ");

outtextxy(140,335," = ");

outtextxy(250,335," . ");

outtextxy(325,167," + ");

outtextxy(380,167," - ");

outtextxy(325,225," / ");

outtextxy(383,228," \* ");

settextstyle(12,0,7);

outtextxy(325,290," SQRT ");

outtextxy(325,300," (s) ");

outtextxy(380,290," REM ");

outtextxy(380,300," (m) ");

outtextxy(325,345," POW ");

outtextxy(325,355," (p) ");

outtextxy(380,345," OFF ");

float x,y,z;

char choice;

setviewport(140,95,420,130,1);

gotoxy(25,7);

cin>>x;

clearviewport();

gotoxy(25,7);

cin>>choice;

if(choice=='s') // s represents operation choice

{

}

else

{

clearviewport();

gotoxy(25,7);

cin>>y;

}

calculate(x,choice,y);

cout<<" Press 'n' to exit";

}

/////////////////////////////////////////////////

//MATRIX MANIPULATOR

void operation()

{

oper:

cleardevice();

setbkcolor(BLACK);

setcolor(GREEN);

int q;

int E[50][50];

int Q[50][50];

int W[50][50];

int A[50][50];

int B[50][50];

int C[50][50];

settextstyle(7,0,3);

outtextxy(60,10," ENTER YOUR CHOICE ");

outtextxy(60,80,"1. Matrix Addition ");

outtextxy(60,160,"2. Matrix Substraction ");

outtextxy(60,240,"3. Matrix Multiplication ");

outtextxy(60,320,"4. Diagonal Operation ");

cin>>q;

switch(q)

{

case 1:

{

cleardevice();

setbkcolor(BLUE);

setcolor(WHITE);

settextstyle(7,0,3);

outtextxy(6,20,"Enter the elements of the 1st matrix ");

int ab=5;

for (int w=0; w<3; w++)

{

for (int e=0; e<3; e++)

{

gotoxy((5\*e)+5,ab);

cin>>A[w][e];

}

ab++;

}

cout<<"\n\n\n\n";

outtextxy(6,120,"Enter the elements of the 2nd matrix ");

int bc=12;

for (int r=0; r<3; r++)

{

for (int t=0; t<3; t++)

{

gotoxy((5\*t)+5,bc);

cin>>B[r][t];

}

bc++;

}

cout<<"\n\n\n\n\n";

for (int y=0; y<3; y++)

{

for (int u=0; u<3; u++)

C[y][u]=A[y][u]+B[y][u];

}

outtextxy(10,250,"Addition of both the matrices is");

for ( r=0; r<3; r++)

{

cout<<endl;

for (int t=0; t<3; t++)

{

cout<<setw(5)<<C[r][t];

}

}

}

break;

case 2 :

{

cleardevice();

setbkcolor(BLUE);

setcolor(WHITE);

settextstyle(7,0,3);

outtextxy(6,20,"Enter the elements of the 1st matrix ");

int cd=5;

for (int w=0; w<3; w++)

{

for (int e=0; e<3; e++)

{

gotoxy((5\*e)+5,cd);

cin>>A[w][e];

}

cd++;

}

cout<<"\n\n\n\n";

outtextxy(6,120,"Enter the elements of the 2nd matrix ");

int de=12;

for (int r=0; r<3; r++)

{

for (int t=0; t<3; t++)

{

gotoxy((5\*t)+5,de);

cin>>B[r][t];

}

de++;

}

cout<<"\n\n\n\n\n";

for (int y=0; y<3; y++)

{

for (int u=0; u<3; u++)

C[y][u]=A[y][u]-B[y][u];

}

outtextxy(10,250,"Substraction of both the matrices is");

for ( r=0; r<3; r++)

{

for (int t=0; t<3; t++)

{

cout<<setw(5)<<C[r][t];

}

cout<<endl;

}

}

break;

case 3:

{

cleardevice();

setbkcolor(BLACK);

setcolor(LIGHTMAGENTA);

settextstyle(7,0,3);

int ef=5;

outtextxy(6,20,"Enter the elements of the 1st matrix ");

for (int w=0; w<3; w++)

{

for (int e=0; e<3; e++)

{

gotoxy((5\*e)+5,ef);

cin>>A[w][e];

}

ef++;

}

cout<<"\n\n\n\n";

outtextxy(6,120,"Enter the elements of the 2nd matrix ");

int fg=12;

for (int r=0; r<3; r++)

{

for (int t=0; t<3; t++)

{

gotoxy((5\*t)+5,fg);

cin>>B[r][t];

}

fg++;

}

cout<<"\n\n\n\n\n";

for (int y=0; y<3; y++)

{

for (int u=0; u<3; u++)

{

C[y][u]=0;

for (int i=0; i<3; i++)

C[y][u]=C[y][u]+(A[y][i]\*B[i][u]);

}

}

outtextxy(10,250,"Multiplication of both the matrices is ");

for ( r=0; r<3; r++)

{

for (int t=0; t<3; t++)

{

cout<<setw(5)<<C[r][t];

}

cout<<endl;

}

}

break;

case 4:

cleardevice();

setbkcolor(DARKGRAY);

setcolor(BLUE);

settextstyle(3,0,3);

int a[30][30];

int b=5,c=0;

int below=0;

int above=0;

outtextxy(6,20,"Enter the elements of the matrix ");

for(int i=0; i<3; i++)

{

for (int j=0; j<3; j++)

{

gotoxy((5\*j)+5,b);

cin>>a[i][j];

}

b++;

}

cout<<"\n\n\n\n";

for ( i=0; i<3; i++)

{

for (int j=0;j<3;j++)

{

if(i==j)

c=c+a[i][j];

}}

cout<<" Trace of the matrix is : "<<c;

// upper and lower diagonal elements sum

for(i=0;i<3;i++)

{

for (int j=0;j<3;j++)

{

if(i>j)

{

below=below+a[i][j];

}

}

}

for(i=0;i<3;i++)

{

for (int j=0;j<3;j++)

{

if(i<j)

{

above=above+a[i][j];

}

}

}

cout<<endl<<endl<<endl;

cout<<" The sum of lower triangular elements is : "<<below;

cout<<endl<<endl<<endl;

cout<<" The sum of upper triangular elements is : "<<above;

break;

}

cout<<"\n\n\n\n";

char y;

cout<<endl<<endl;

cout<<"Press 'y' for another operation"<<endl;

cout<<"Press 'n' to exit "<<endl;

cin>>y;

if(y=='y')

goto oper;

else exit(0);

}

///////////////////////////////////

//transpose of matrix

void transpose()

{

cleardevice();

setbkcolor(BLACK);

setcolor(GREEN);

settextstyle(7,0,3);

int a[30][30],b[30][30];

int c=5;

outtextxy(10,20,"Enter the elements of the matrix");

for (int i=0; i<3; i++)

{

for (int j=0; j<3; j++)

{

gotoxy((5\*j)+5,c);

cin>>a[i][j];

}

c++;

}

for (int k=0; k<3; k++)

{

for (int c=0; c<3; c++)

b[k][c]=a[c][k];

}

cout<<endl<<endl<<endl;

outtextxy(6,150,"The transpose of the matrix is");

for (k=0;k<3;k++)

{

cout<<endl;

for (int c=0;c<3;c++)

{

cout<<setw(5)<<b[k][c];

}

cout<<endl;

}

}

/////////////////////////////////////////////

void matrix()

{

cleardevice();

setbkcolor(BLACK);

setcolor(LIGHTRED);

int choice1;

settextstyle(1,0,4);

outtextxy(300,10," INDEX :");

cout<<"\n\n\n\n";

outtextxy(60,60,"1.Operations on Matrices ");

outtextxy(60,140,"2. Transpose of a Matrix ");

cout<<endl<<endl<<endl<<endl<<"\t\t\t\t\t\t\t\t\t\t";

cin>>choice1;

switch(choice1)

{

case 1:operation();

break;

case 2:transpose();

break;

default:cout<<endl<<endl<<endl<<endl<<endl<<endl<<"\t\t\t\t SORRY! Wrong choice";

break;

}

}

//////////////////////////////////////////////////////////

//FRONT PAGE

void front()

{

int gd=DETECT,gm,i,a;

initgraph(&gd,&gm, "C:\\TURBOC3\\BGI");

cleardevice();

setviewport(0,0,639,479,1);

clearviewport();

cleardevice();

setbkcolor(WHITE);

setcolor(RED);

settextstyle(7,0,3);

outtextxy(260,150,"C.S PROJECT ");

outtextxy(60,190,"Calculator and Matrix Manipulator ");

outtextxy(60,340,"Project By: ");

outtextxy(240,340,"Aashita Gloria Noah ");

//delay(4000);

}

///////////////////////////////////////////////

class stud

{

public:

char name[50],remarks[100];

int age;

void input()

{

cout<<"Enter your name "<<endl;

gets(name);

cout<<"enter your age "<<endl;

cin>>age;

}

void iremarks()

{

cout<<"Your Valuable Remarks "<<endl;

gets(remarks);

}

void display()

{

cout<<"Name : "<<name<<endl;

cout<<"Age : "<<age<<endl;

cout<<"Your Valuable Remarks :"<<remarks<<endl;

}

}obj;

///////////////////////////////////////////////

//MAIN MENU

void main()

{

clrscr();

fstream file;

file.open("remarks.dat", ios::in|ios::out|ios::app|ios::binary);

obj.input();

file.write((char\*)&obj,sizeof(obj));

file.close();

front();

int gd=DETECT,gm,i,a;

initgraph(&gd,&gm, "C:\\TURBOC3\\BGI");

setviewport(0,0,639,479,1);

clearviewport();

int choice;

setbkcolor(BLACK);

setcolor(MAGENTA);

settextstyle(7,0,3);

outtextxy(150,5," MAIN MENU ");

outtextxy(60,80," 1. Calculator");

outtextxy(60,160," 2. Matrix Manipulator");

outtextxy(60,240," 3. Exit");

cout<<endl<<endl<<endl<<endl<<endl<<endl<<endl<<endl<<endl<<endl<<endl<<endl<<endl<<endl<<endl<<endl<<endl<<endl<<endl<<"\t\t\t\tEnter Your Choice ";

cin>>choice;

int A,B;

switch(choice)

{

case 1:calculator();

break;

case 2:matrix();

break;

case 3: exit(0);

default:cout<<"\n\t\t\t\tSorry! Wrong choice.";

break;

}

/////////////////////////////////////////////////////

////binary file

getch();

cleardevice();

int s;

cout<<"REMARK SCREEN"<<endl<<endl<<endl;

do

{

cout<<" 1.INPUT REMARKS DETAILS"<<endl;

cout<<" 2.DISPLAY DETAILS"<<endl;

cout<<" 3.EXIT"<<endl<<endl;

cout<<"ENTER YOUR CHOICE"<<endl;

cin>>s;

switch(s)

{

case 1:

file.open("remarks.dat", ios::in|ios::out|ios::app|ios::binary);

obj.iremarks();

file.write((char\*)&obj,sizeof(obj));

file.close();

cout<<endl;

break;

case 2:

file.open("remarks.dat", ios::in|ios::out|ios::app|ios::binary);

file.seekg(0);

while(file.read((char\*)&obj,sizeof(obj)))

{

obj.display();

}

file.close();

cout<<endl;

break;

case 3:

file.open("remarks.dat", ios::in|ios::out|ios::app|ios::binary);

exit(0);

file.close();

cout<<endl;

break;

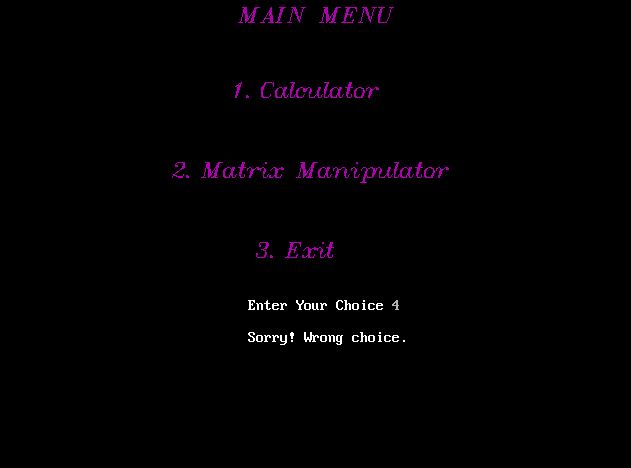
}

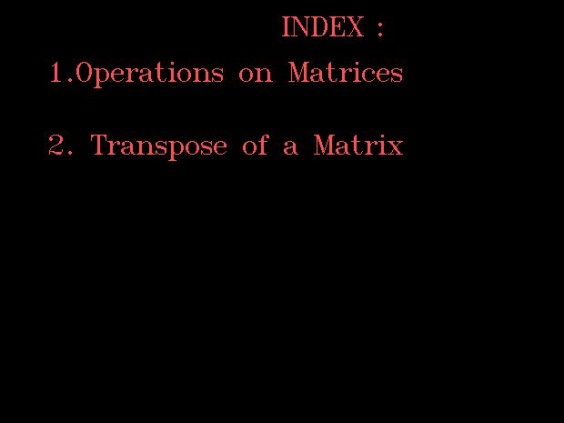
}while(s<4);

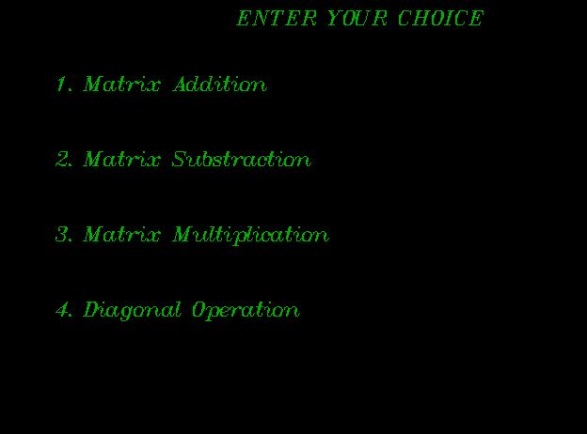
getch();

}

**SNAPSHOTS:**







**CONCLUSION**

This project is a humble venture which allows the user to perform his desired operations on a calculator and matrix and permits the user to explore the different operations which can be performed on a matrix.