Tribhuvan University

Institute of Engineering

Pulchowk Campus

PROJECT WORK ON C PROGRAMMING

Submitted by:

Rachana Subedi (076BEL033)

Roshan Niroula(076BEL035)

Shashwot Shrestha(076BEL043)

Sushil Phuyal(076BEL047)

Swodesh Sharma(076BEL048)

Submitted to :

Department of Electronics

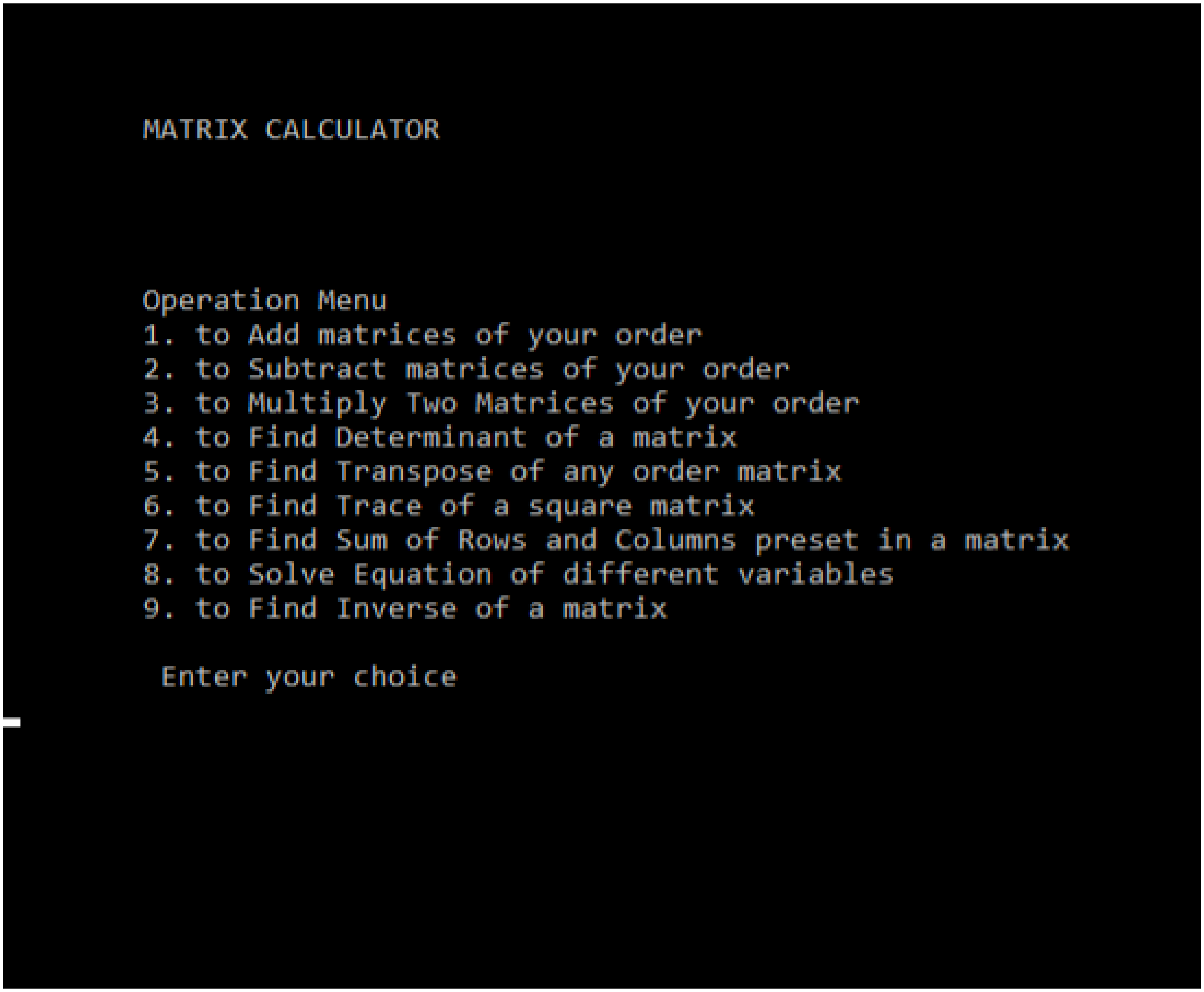
and Computer Engineering

(Kumar Prasun sir)

Project:

Matrix Calculator

Output Samles:

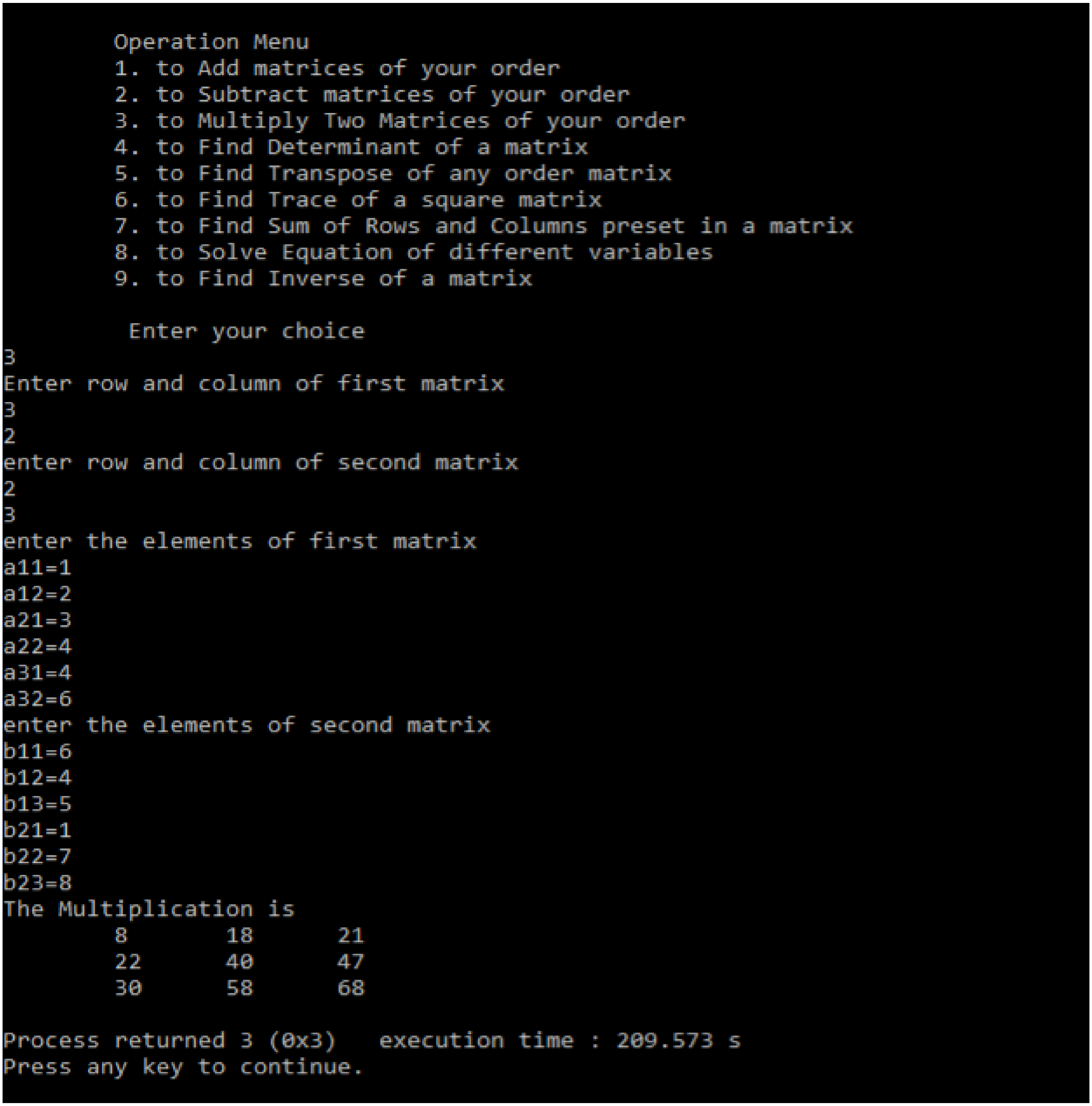


```
      MATRIX CALCULATOR




      Operation Menu
      1. to Add matrices of your order
      2. to Subtract matrices of your order
      3. to Multiply Two Matrices of your order
      4. to Find Determinant of a matrix
      5. to Find Transpose of any order matrix
      6. to Find Trace of a square matrix
      7. to Find Sum of Rows and Columns preset in a matrix
      8. to Solve Equation of different variables
      9. to Find Inverse of a matrix

       Enter your choice
_
```



```
        Operation Menu
        1. to Add matrices of your order
        2. to Subtract matrices of your order
        3. to Multiply Two Matrices of your order
        4. to Find Determinant of a matrix
        5. to Find Transpose of any order matrix
        6. to Find Trace of a square matrix
        7. to Find Sum of Rows and Columns preset in a matrix
        8. to Solve Equation of different variables
        9. to Find Inverse of a matrix

         Enter your choice
3
Enter row and column of first matrix
3
2
enter row and column of second matrix
2
3
enter the elements of first matrix
a11=1
a12=2
a21=3
a22=4
a31=4
a32=6
enter the elements of second matrix
b11=6
b12=4
b13=5
b21=1
b22=7
b23=8
The Multiplication is
        8       18      21
        22      40      47
        30      58      68

Process returned 3 (0x3)   execution time : 209.573 s
Press any key to continue.
```

and many more...

Source Code:-

```
#include<stdio.h>
#include<conio.h>
#include<math.h>

int main()
{
    int i,j,k,m,n,p,q,d,sum;
    int a[10][10],b[10][10],c[10][10];
    float inverse[3][3],inv[3][3],mat[3][3];
    int sumr,sumc,s=0;
    signed int a1,b1,c1,d1;
    signed int a2,b2,c2,d2;
    signed int a3,b3,c3,d3;
    float x,y,z;
    float A1,B1,D1;
    float A2,B2,D2;
    int operation;
    printf("\n\n\n\nMATRIX CALCULATOR\n\n\n\n");
    printf("\nOperation Menu\n");
    printf("\t1. to Add matrices of your order \n");
    printf("\t2. to Subtract matrices of your order\n");
    printf("\t3. to Multiply Two Matrices of your order\n");
    printf("\t4. to Find Determinant of a matrix\n");
    printf("\t5. to Find Transpose of any order matrix \n");
    printf("\t6. to Find Trace of a square matrix\n");
    printf("\t7. to Find Sum of Rows and Columns preset in a matrix\n");
    printf("\t8. to Solve Equation of different variables\n");
    printf("\t9. to Find Inverse of a matrix\n");
    printf("\n Enter your choice \n");
    scanf("%d",&operation);
    switch(operation){


case 1:

    printf("\n Enter size of matrix \n ");
    scanf("%d%d",&m,&n);
    printf("Enter elements in matrix A : \n");
    for(i=0;i<m;i++)
    {
```

```
   for(j=0; j<n;j++)
   {
      printf("a%d%d=",i+1,j+1);
      scanf("%d", &a[i][j]);
   }
   }
  printf("\nEnter elements in matrix B : \n");
  for(i=0; i<m; i++)
  {
  for(j=0; j<n;j++)
  {
      printf("b%d%d=",i+1,j+1);
      scanf("%d", &b[i][j]);
  }
  }
 printf("\nSum of two matrices A+B = \n");
 for(i=0;i<m;i++)
 {
  for(j=0; j<n;j++)
  {
      printf("%d ",(a[i][j] + b[i][j]));
  }
  printf("\n");
}
break;
```

```
case 2:
    printf("\n Enter size of matrix \n ");
    scanf("%d%d",&m,&n);
    printf("Enter elements in matrix A : \n");
    for(i=0;i<m;i++)
    {
    for(j=0; j<n;j++)
    {
      printf("a%d%d=",i+1,j+1);
      scanf("%d", &a[i][j]);
    }
    }
    printf("\nEnter elements in matrix B : \n");
    for(i=0; i<m; i++)
    {
    for(j=0; j<n;j++)
    {
      printf("b%d%d=",i+1,j+1);
      scanf("%d", &b[i][j]);
    }
    }
    printf("\nDifference of two matrices A-B = \n");
    for(i=0; i<m; i++)
    {
    for(j=0; j<n; j++)
    {
      printf("%d ",(a[i][j] - b[i][j]));
    }
    printf("\n");
    }
    break;

case 3:
   printf("Enter row and column of first matrix\n");
   scanf("%d%d",&m,&n);
   printf("enter row and column of second matrix\n");
   scanf("%d%d",&p,&q);
   if(n==p)
   {
   printf("enter the elements of first matrix\n");
   for(i=0;i<m;i++)
    {
    for(j=0;j<n;j++)
    {
      printf("a%d%d=",i+1,j+1);
      scanf("%d",&a[i][j]);
    }
    }
```

```
printf("enter the elements of second matrix\n");
for(i=0;i<p;i++)
 {
  for(j=0;j<q;j++)
 {
    printf("b%d%d=",i+1,j+1);
    scanf("%d",&b[i][j]);

 }


}
for(i=0;i<m;i++)
 {
   for(j=0;j<q;j++)
   {
     c[i][j]=0;
     for(k=0;k<p;k++)
     {

        c[i][j]=c[i][j]+a[i][k]*b[k][j];
     }

   }
 }
```

```
   }

  else{
  printf("not valid");
  }
  printf("The Multiplication is \n");
  for(i=0;i<m;i++)
  {
  for(j=0;j<q;j++)
  {

    printf("\t%d",c[i][j]);
  }
  printf("\n");


  }
  break;




case 4:
  printf("Enter the order of matrix");
   scanf("%d",&k);
```

```
    if(k>10)
        exit(0);
    printf("Enter the matrix of order %d \n",k);
    for(i=0;i<=k-1;i++)
    {
     for(j=0;j<=k-1;j++)
     {
        printf("a%d%d=",i+1,j+1);
        scanf("%d",&a[i][j]);

     }
    }
     d=determinant(a,k);
     printf("Determinant = %d ",d);
     break;

case 5:

     printf("Enter rows and columns: ");
     scanf("%d %d", &m, &n);
     printf("\nEnter matrix elements:\n");
      for (i=0;i<m;i++)
     for (j=0;j<n;j++) {
        printf("Enter element a%d%d: ",i+1,j+1);
        scanf("%d", &a[i][j]);
```

```
 }
for(i=0; i<m; i++)
 {
 for(j=0; j<n; j++)
 {
   printf("%d ",a[i][j]);
 }
 printf("\n");
 }

 for (i=0;i<m;i++)
for (j=0;j<n;j++) {
   c[j][i] = a[i][j];
}



 printf("\nTranspose of the matrix:\n");
for(i=0; i<m; i++)
{
for(j=0; j<n; j++)
{
   printf("%d ",c[i][j]);
}
printf("\n");
}
```

```
     break;


case 6:


    printf("Enter a number of rows and columns:-");
    scanf("%d%d",&m,&n);
    if(m==n)
    {
    printf("\nEnter matrix elements :");
    for(i=0;i<m;i++){
    for(j=0;j<n;j++)
    {
    printf("Enter element a%d%d:", i+1,j+1);
    scanf("%d",&a[i][j]);
    }
    }
    sum=0;
     for(i=0;i<m;i++)
     sum=sum+a[i][i];
    printf("\nTrace of the matrix = %d",sum);
    }
    else
    printf("Not a square matrix. It is not possible to find trace.");
    break;
case 7:


    printf("Enter number of rows and columns of a matrix\n");
    scanf("%d %d",&m,&n);

   printf("Enter Matrix 1\n");
   for(i=0;i<m;i++)
   {
      for(j=0;j<n;j++)
     {
     printf("Enter element a%d%d: ",i+1,j+1);
    scanf("%d",&a[i][j]);
     }
   }

   printf("Given matrix is\n");
       for(i=0;i<m;i++)
        {
        for(j=0;j<n;j++)
        {
          printf("%d\t",a[i][j]);
          }

          printf("\n");
```

```
         }



for(i=0;i<m;i++)
{
 sumr=0;

 for(j=0;j<n;j++)
 {
  sumr+=a[i][j];
 }
 printf("Sum of %d Row is %d\n",s+1,sumr);
    s++;
}
s=0;
 for(i=0;i<n;i++)
 {
   sumc=0;
  for(j=0;j<m;j++)
  {
   sumc+=a[j][i];
   }
     printf("Sum of %d Column is %d\n",s+1,sumc);
     s++;
 }
```

```
  break;

case 8:




printf("\n Choose the format of your equation \n");
printf("\n");
printf("1.Ax+By+Cz=D\n");
printf("\n");
printf("2.Ax+By+Cz=0\n");
printf("\n");
printf("3.Ax+By=D\n");
printf("\n");
printf("4.Ax+By=0\n");
printf("\n");
printf("Your choice =");
scanf("%d",&n);
switch(n)
{
case 1:
printf("\n Enter coeffiecients of the x,y,z and the value of d of first equation\n");
scanf("%d%d%d%d",&a1,&b1,&c1,&d1);
```

```
printf("\n Enter coeffiecients of the x,y,z and the value of d of second equation\n");
scanf("%d%d%d%d",&a2,&b2,&c2,&d2);
printf("\n Enter coeffiecients of the x,y,z and the value of d of third equation\n");
scanf("%d%d%d%d",&a3,&b3,&c3,&d3);

A1=(a1*c2)-(a2*c1);
B1=(b1*c2)-(b2*c1);
D1=(d1*c2)-(d2*c1);
A2=(a2*c3)-(a3*c2);
B2=(b2*c3)-(b3*c2);
D2=(d2*c3)-(d3*c2);

x=(float)((D1*B2)-(D2*B1))/((A1*B2)-(A2*B1));
y=(float)((D1*A2)-(D2*A1))/((A2*B1)-(A1*B2));
z=(((float)d1)-(a1*x)-(b1*y))/c1;

printf("\n The equations are\n");
printf(" %dx + %dy + %dz = %d ........(1)\n",a1,b1,c1,d1);
printf(" %dx + %dy + %dz = %d ........(2)\n",a2,b2,c2,d2);
printf(" %dx + %dy + %dz = %d ........(3)\n",a3,b3,c3,d3);
printf("The value of x=%.3f\ty=%.3f\tz=%.3f",x,y,z);

break;
```

```
case 2:

printf("\n Enter coeffiecients of the x,y,z of first equation\n");
scanf("%d%d%d",&a1,&b1,&c1);
printf("\n Enter coeffiecients of the x,y,z of second equation\n");
scanf("%d%d%d",&a2,&b2,&c2);
printf("\n Enter coeffiecients of the x,y,z of third equation\n");
scanf("%d%d%d",&a3,&b3,&c3);
d1=d2=d3=0;
A1=(a1*c2)-(a2*c1);
B1=(b1*c2)-(b2*c1);
D1=(d1*c2)-(d2*c1);
A2=(a2*c3)-(a3*c2);
B2=(b2*c3)-(b3*c2);
D2=(d2*c3)-(d3*c2);

x=(float)((D1*B2)-(D2*B1))/((A1*B2)-(A2*B1));
y=(float)((D1*A2)-(D2*A1))/((A2*B1)-(A1*B2));
z=(((float)d1)-(a1*x)-(b1*y))/c1;

printf("\n The equations are\n");
printf(" %dx + %dy + %dz = %d ........(1)\n",a1,b1,c1,d1);
printf(" %dx + %dy + %dz = %d ........(2)\n",a2,b2,c2,d2);
printf(" %dx + %dy + %dz = %d ........(3)\n",a3,b3,c3,d3);
printf("The value of x=%.3f\ty=%.3f\tz=%.3f",x,y,z);

break;

case 3:
printf("\n Enter coeffiecients of the x,y the value of d of first equation\n");
scanf("%d%d%d",&a1,&b1,&d1);
printf("\n Enter coeffiecients of the x,y and the value of d of second equation\n");
scanf("%d%d%d",&a2,&b2,&d2);

x=(float)((d1*b2)-(d2*b1))/((a1*b2)-(a2*b1));
y=(float)((d1*a2)-(d2*A1))/((a2*b1)-(a1*b2));

printf("\n The equations are\n");
printf(" %dx + %dy = %d ........(1)\n",a1,b1,d1);
printf(" %dx + %dy = %d ........(2)\n",a2,b2,d2);
printf("The value of x=%.3f\ty=%.3f",x,y);
break;

case 4:
printf("\n Enter coeffiecients of the x,y of first equation\n");
scanf("%d%d",&a1,&b1);
printf("\n Enter coeffiecients of the x,y of second equation\n");
scanf("%d%d",&a2,&b2);
```

```
d1=d2=0;
x=(float)((d1*b2)-(d2*b1))/((a1*b2)-(a2*b1));
y=(float)((d1*a2)-(d2*A1))/((a2*b1)-(a1*b2));

printf("\n The equations are\n");
printf(" %dx + %dy = %d ........(1)\n",a1,b1,d1);
printf(" %dx + %dy = %d ........(2)\n",a2,b2,d2);
printf("The value of x=%.3f\ty=%.3f",x,y);
break;

}
break;

case 9:
    printf("\t1. to Find 3*3 matrix\n");
    printf("\t2. to Find 2*2 matrix\n");

    printf("\n Enter your choice \n");
     scanf("%d",&d);
     switch(d){
    case 1:
    printf("enter elements of 3*3 matrix:");
   for (i=0;i<3;i++)
   {
   for (j=0;j<3;j++)
```

```
 {
  printf("a%d%d=",i+1,j+1);
  scanf("%f",&mat[i][j]);
 }
 }
 printf("given matrix is: \n");
 for (i=0;i<3;i++)
 {
 for (j=0;j<3;j++)
 {
 printf("%.2f \t",mat[i][j]);
 }
 printf("\n");
 }
 printf("\n");
 printf("\n");
 d=(mat[0][0]*((mat[1][1]*mat[2][2])-(mat[1][2]*mat[2][1])))-
(mat[0][1]*((mat[1][0]*mat[2][2])-
(mat[1][2]*mat[2][0])))+(mat[0][2]*((mat[1][0]*mat[2][1])-(mat[1][1]*mat[2][0])));
  printf("determinant is %d \n",d);
  printf("\n");
 printf("inversed is; \n");
  for (i=0;i<3;i++)
 {
 for (j=0;j<3;j++)
```

```
  {
        if(i==0 && j==0)
        inv[i][j]=((mat[1][1]*mat[2][2])-(mat[1][2]*mat[2][1]))/d;
        if ( i==0 && j==1)
        inv[i][j]=-((mat[1][0]*mat[2][2])-(mat[1][2]*mat[2][0]))/d;
        if ( i==0 && j==2)
        inv[i][j]=((mat[1][0]*mat[2][1])-(mat[1][1]*mat[2][0]))/d;
        if ( i==1 && j==0)
        inv[i][j]=-((mat[0][1]*mat[2][2])-(mat[0][2]*mat[2][1]))/d;
        if ( i==1 && j==1)
        inv[i][j]=((mat[0][0]*mat[2][2])-(mat[0][2]*mat[2][0]))/d;
        if ( i==1 && j==2)
        inv[i][j]=-((mat[0][0]*mat[2][1])-(mat[0][1]*mat[2][0]))/d;
        if ( i==2 && j==0)
        inv[i][j]=((mat[0][1]*mat[1][2])-(mat[0][2]*mat[1][1]))/d;
        if ( i==2 && j==1)
        inv[i][j]=-((mat[0][0]*mat[1][2])-(mat[0][2]*mat[1][0]))/d;
        if ( i==2 && j==2)
        inv[i][j]=((mat[0][0]*mat[1][1])-(mat[0][1]*mat[1][0]))/d;

}
}
for (i=0;i<3;i++)
  {
  for (j=0;j<3;j++)
  {
    if (i==j)
      inverse[i][j]=inv[i][j];
    else
  inverse[i][j]=inv[j][i];
  }
  }


for (i=0;i<3;i++)
  {
  for (j=0;j<3;j++)
  {
  printf("%.2f\t",inverse[i][j]);
  }
  printf("\n");
}
break;
case 2:
   printf("enter elements of 2*2 matrix:");
  for (i=0;i<2;i++)
  {
  for (j=0;j<2;j++)
  {
    printf("a%d%d=",i+1,j+1);
```

```
      scanf("%f",&mat[i][j]);
    }
    }
    printf("given matrix is: \n");
    for (i=0;i<2;i++)
    {
    for (j=0;j<2;j++)
    {
    printf("%.2f \t",mat[i][j]);
    }
    printf("\n");
    }
    printf("\n");
    printf("\n");
    d=mat[0][0]*mat[1][1]-mat[0][1]*mat[1][0];
     printf("determinant is %d \n",d);
     printf("\n");
    printf("inversed is; \n");

  for (i=0;i<2;i++)
    {
    for (j=0;j<2;j++)
    {
      if (i==j)
      {
```

```
      if(i==0)
      inverse[i][j]=mat[1][1]/d;
      else
      inverse[i][j]=mat[0][0]/d;
    }
      else
  inverse[i][j]=-mat[i][j]/d;
  }
  }


for (i=0;i<2;i++)
  {
  for (j=0;j<2;j++)
  {
  printf("%.2f\t",inverse[i][j]);
  }
  printf("\n");
}
break;
    }
    break;
```

```
     }
}
int determinant(int a[10][10], int k)
{
    int det,c,s=1,b[10][10],i,j,m,n;
    if(k==1)
        return(a[0][0]);
    else{
        det=0;
         for(c=0;c<=k-1;c++){
         m=0,n=0;
          for(i=0;i<k;i++){
            for(j=0;j<k;j++){
                b[i][j]=0;
                if(i!=0&&j!=c){
                  b[m][n]=a[i][j];
                  if(n<(k-2))
                    n++;
                else{
                    n=0;
                    m++;
                }
              }
```

```
      }
    }
    det=det+s*(a[0][c]*determinant(b,k-1));
    s=-1*s;
   }

  }
  return(det);
}
```

The END..