1. 2D

```
#include<conio.h>
#include<iostream.h>
#include<math.h>
#include<graphics.h>
#define PI 3.14

void translation(int a[10][10],int n);
void scaling(int a[10][10],int n);
void rotation(int a[10][10],int n);
void shearing(int a[10][10],int n);
void display(int a[10][10],int mul[10][10],int n);
void main()
{
  clrscr();
  int a[10][10],n,choice;
  cout<<"Enter the no. of vertices:";
   cin>>n;
  cout<<"Enter the vertices:";
  for(int i = 0;i<n;i++)
   {
        for(int j = 0;j<3;j++)
         {
           if(j==2)
             a[i][j]=1;
           else
             cin>>a[i][j];
         }
   }

  cout<<"Enter the operation u want to perform :\n";
  cout<<"1. Translation\n";
  cout<<"2. Scaling\n";
  cout<<"3. Rotation\n";
  cout<<"4. Shearing\n";
  cout<<"Enter the operation u want to perform : ";
  cin>>choice;

  switch(choice)
   {
        case 1: translation(a,n);
                  break;
        case 2: scaling(a,n);
                  break;
```

```
        case 3: rotation(a,n);
                break;
        case 4: shearing(a,n);
                break;
        default: cout<<"Invalid input";
    }
  getch();
}
void translation(int a[10][10],int n)
{
    int arr[3][3],mul[10][10],trans[10],k=0;
    cout<<"Enter the translation along x-axis:";
    cin>>trans[0];
    cout<<"Enter the translation along y-axis:";
    cin>>trans[1];

    for(int i = 0;i<3;i++)
    {
        for(int j = 0;j<3;j++)
        {
           arr[i][j]=0;
           if(i==j)
           {
            arr[i][j]=1;
            k++;
           }
        }
    }
    arr[2][0]=trans[0];
    arr[2][1]=trans[1];

    for(i = 0;i<n;i++)
    {
        for(int j = 0;j<3;j++)
        {
           mul[i][j]= 0;
           for(k= 0;k<3;k++)
           {
                mul[i][j]+=a[i][k]*arr[k][j];
           }
        }
     }
     display(a,mul,n);
    /*for(i = 0;i<n;i++)
```

```
    {
        for(int j = 0;j<3;j++)
         {
            cout<<mul[i][j]<<" ";
         }
        cout<<"\n";
    } */

}
void scaling(int a[10][10],int n)
{
   int arr[3][3],mul[10][10],scale[10],k=0;
   cout<<"Enter the scaling factor along x-axis:";
   cin>>scale[0];
   cout<<"Enter the scaling factor along y-axis:";
   cin>>scale[1];

   for(int i = 0;i<3;i++)
   {
                for(int j = 0;j<3;j++)
                {
            arr[i][j]=0;
            if(i==j)
             {
              arr[i][j]=scale[k];
              k++;
             }
                }
   }
   arr[2][2]=1;
   for(i = 0;i<n;i++)
   {
                for(int j = 0;j<3;j++)
                {
                        mul[i][j]= 0;
                        for(k= 0;k<3;k++)
                {
                                mul[i][j]+=a[i][k]*arr[k][j];
                }
                }
   }
   display(a,mul,n);
  /* for(i = 0;i<n;i++)
    {
```

```
            for(int j = 0;j<3;j++)
              {
                 cout<<mul[i][j]<<" ";
              }
            cout<<"\n";
      }     */
}
void shearing(int a[10][10],int n)
{
    int arr[3][3]={0},mul[10][10],shear[10],k=0;
    cout<<"Enter the shearing in x-direction:";
    cin>>shear[0];
    cout<<"Enter the shearing in y-direction:";
    cin>>shear[1];
    arr[0][0]=arr[1][1]=1;
    arr[0][1]=shear[0];
    arr[2][2]=1;
    arr[1][0]=shear[1];
    for(int i = 0;i<n;i++)
     {
           for(int j = 0;j<3;j++)
            {
              mul[i][j]= 0;
              for(k= 0;k<3;k++)
               {
                     mul[i][j]+=a[i][k]*arr[k][j];
               }
             }
      }
    for(i = 0;i<n;i++)
      {
            for(int j = 0;j<3;j++)
              {
                 cout<<mul[i][j]<<" ";
              }
            cout<<"\n";
      }
     display(a,mul,n);

}
void rotation(int a[10][10],int n)
{
    int theta,i,j,k= 0;
    float arr[3][3],mul[10][10],val=PI/180,ag,bg;
```

```
int dir;

cout<<"Enter the direction of rotation:\n";
cout<<"Press 1 for anticlockwise\n";
cout<<"Press 2 for clockwise";
cin>>dir;
cout<<"Enter the angle for rotation:";
cin>>theta;
ag=sin(theta*val);
bg=cos(theta*val);
if(dir==1)
	{
	arr[0][0]=arr[1][1]=bg;
	arr[0][1]=-ag;
	arr[1][0]=ag;
	arr[2][0]=arr[2][1]=arr[0][2]=arr[1][2]=0;
	arr[2][2]=1;
}
else
{
	arr[0][0]=arr[1][1]=bg;
	arr[0][1]=ag;
	arr[1][0]=-ag;
	arr[2][0]=arr[2][1]=arr[0][2]=arr[1][2]=0;
	arr[2][2]=1;
}

for(i = 0;i<n;i++)
 {
	for(int j = 0;j<3;j++)
	 {
	   mul[i][j]= 0;
	   for(k= 0;k<3;k++)
	    {
		mul[i][j]+=a[i][k]*arr[k][j];
	    }
	 }
  }
for(i = 0;i<n;i++)
  {
	 for(int j = 0;j<3;j++)
	   {
	      cout<<mul[i][j]<<" ";
	   }
```

```
            cout<<"\n";
      }
   //  display(a,mul,n);
}
void display(int a[10][10],int mul[10][10],int n)
{
    int gd = DETECT, gm;
    initgraph (&gd, &gm, "C:\\TURBOC3\\BGI");
    if(n==2)
     {
       line(a[0][0],a[0][1],a[1][0],a[1][1]);
       line(mul[0][0],mul[0][1],mul[1][0],mul[1][1]);
     }
    else
     {
       for(int i=0;i<(n-1);i++)
            {
              line(a[i][0],a[i][1],a[i+1][0],a[i+1][1]);
              line(mul[i][0],mul[i][1],mul[i+1][0],mul[i+1][1]);
            }
              line(a[0][0],a[0][1],a[n-1][0],a[n-1][1]);
              line(mul[0][0],mul[0][1],mul[n-1][0],mul[n-1][1]);

     }
    getch();
    closegraph();
    getch();
}
```