DATE: 19/10/2023

GROUP B: PRATICAL.4

NAME: SAYALI TANAJI PAWAR
ROLL NO: S213002
CLASS: SE
DIV: C
BATCH: C1

PROBLEM STATEMENT:

Write C++ program to draw 2-D object and perform following basic transformations,a) Scaling b) Translation c) Rotation. Apply concept of operator overloading.

CODE:

```
#include<iostream>
#include<graphics.h>
#include<math.h>
using namespace std;
class transform                          //creating class for functions
{
public:
int m,a[20][20],c[20][20];
int i,j,k;
public:
void object();
void accept();
void operator *(float b[20][20])          //overloading * operator
{
for(int i=0;i<m;i++)
{
for(int j=0;j<m;j++)
{
c[i][j]=0;
for(int k=0;k<m;k++)
{
c[i][j]=c[i][j]+(a[i][k]*b[k][j]);
}
}
```

```
}
}
};
void transform::object()          //for setting the x y axis on polygon coordinates
{
int gd,gm;
gd=DETECT;                              //graphics initialization
initgraph(&gd,&gm,NULL);
line(300,0,300,600);                        //x axis
line(0,300,600,300);                        //y axis
for( i=0;i<m-1;i++)
{
line(300+a[i][0],300-a[i][1],300+a[i+1][0],300-a[i+1][1]);
}
line(300+a[0][0],300-a[0][1],300+a[i][0],300-a[i][1]);
for( i=0;i<m-1;i++)
{
line(300+c[i][0],300-c[i][1],300+c[i+1][0],300-c[i+1][1]);
}
line(300+c[0][0],300-c[0][1],300+c[i][0],300-c[i][1]);
int temp;
cout << "Press 1 to continue";
cin >> temp;
closegraph();
}
void transform::accept()              //function for accepting polygon info
{
cout<<"\n";
cout<<"Enter the Number Of Edges:";
cin>>m;
cout<<"\nEnter The Coordinates :";
for(int i=0;i<m;i++)
{
for(int j=0;j<3;j++)
{
if(j>=2)
a[i][j]=1;
else
cin>>a[i][j];
}
}
}
int main()
{
int ch,tx,ty,sx,sy;
```

```
float deg,theta,b[20][20];
transform t;
t.accept();
cout<<"\nEnter your choice";
cout<<"\n1.Translation"
"\n2.Scaling"
"\n3.Rotation";
cin>>ch;
switch(ch)
{
case 1: cout<<"\nTRANSLATION OPERATION\n";
cout<<"Enter value for tx and ty:";
cin>>tx>>ty;
b[0][0]=b[2][2]=b[1][1]=1;      //setting matrix elements according to translation
b[0][1]=b[0][2]=b[1][0]=b[1][2]=0;
b[2][0]=tx;
b[2][1]=ty;
t * b;
t.object();
break;
case 2: cout<<"\nSCALING OPERATION\n";
cout<<"Enter value for sx,sy:";
cin>>sx>>sy;
b[0][0]=sx;                    //setting matrix elements according to scaling
b[1][1]=sy;
b[0][1]=b[0][2]=b[1][0]=b[1][2]=0;
b[2][0]=b[2][1]=0;
b[2][2] = 1;
t * b;
t.object();
break;
case 3: cout<<"\nROTATION OPERATION\n";
cout<<"Enter value for angle:";
cin>>deg;
theta=deg*(3.14/100);
b[0][0]=b[1][1]=cos(theta);      //setting matrix elements according to rotation
b[0][1]=sin(theta);
b[1][0]=sin(-theta);
b[0][2]=b[1][2]=b[2][0]=b[2][1]=0;
b[2][2]=1;
t * b;
t.object();
break;
default:
cout<<"\nInvalid choice";
```

```
}
getch();
return 0;
}
```

INPUT :

1.Translation:

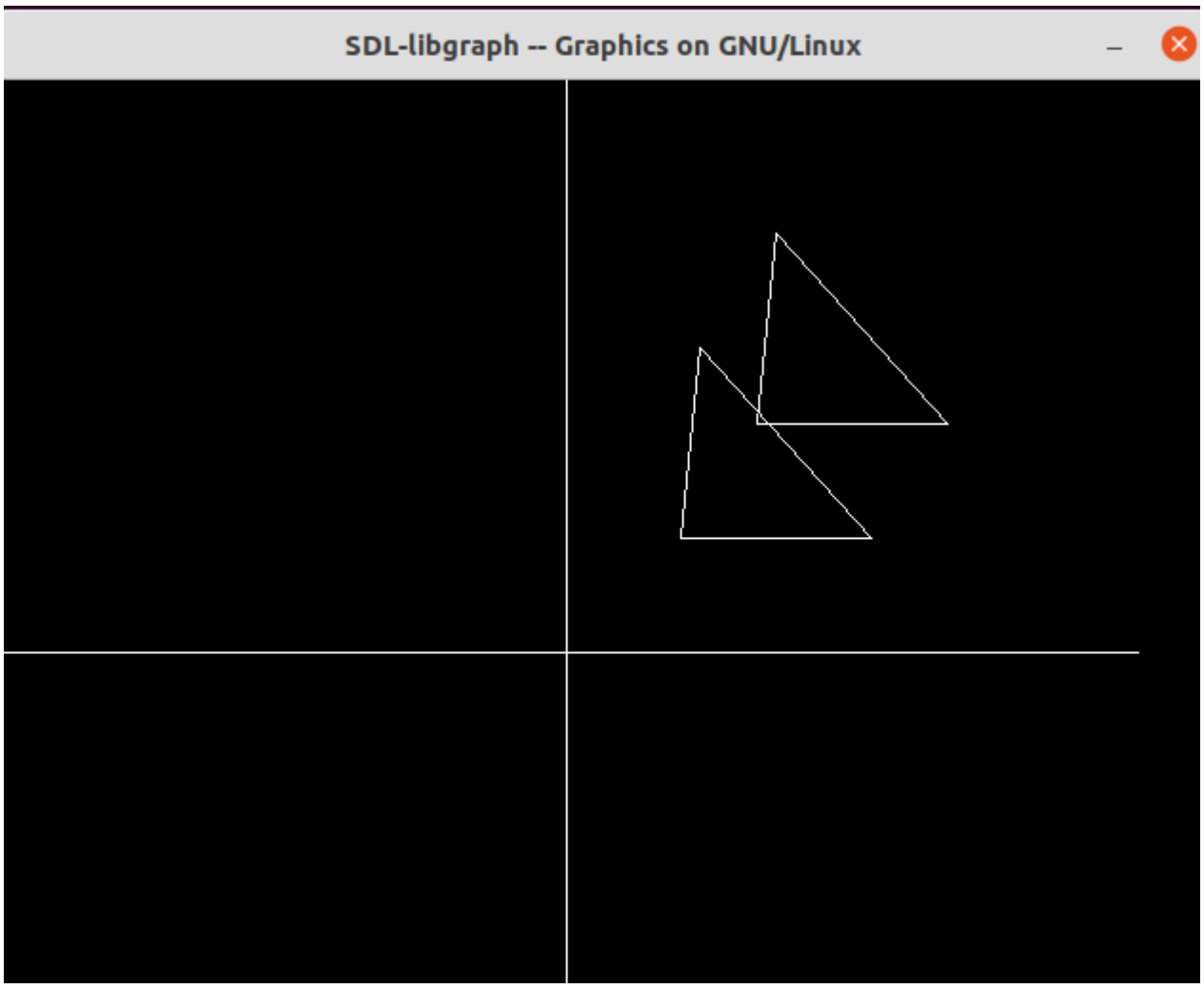
```
Enter the Number Of Edges: 3

Enter The Coordinates :60
60
160
60
70
160

Enter your choice
1.Translation
2.Scaling
3.Rotation 1

TRANSLATION OPERATION
Enter value for tx and ty:40
60
```

OUTPUT:



INPUT:

2.Scaling:

Enter the Number Of Edges: 3

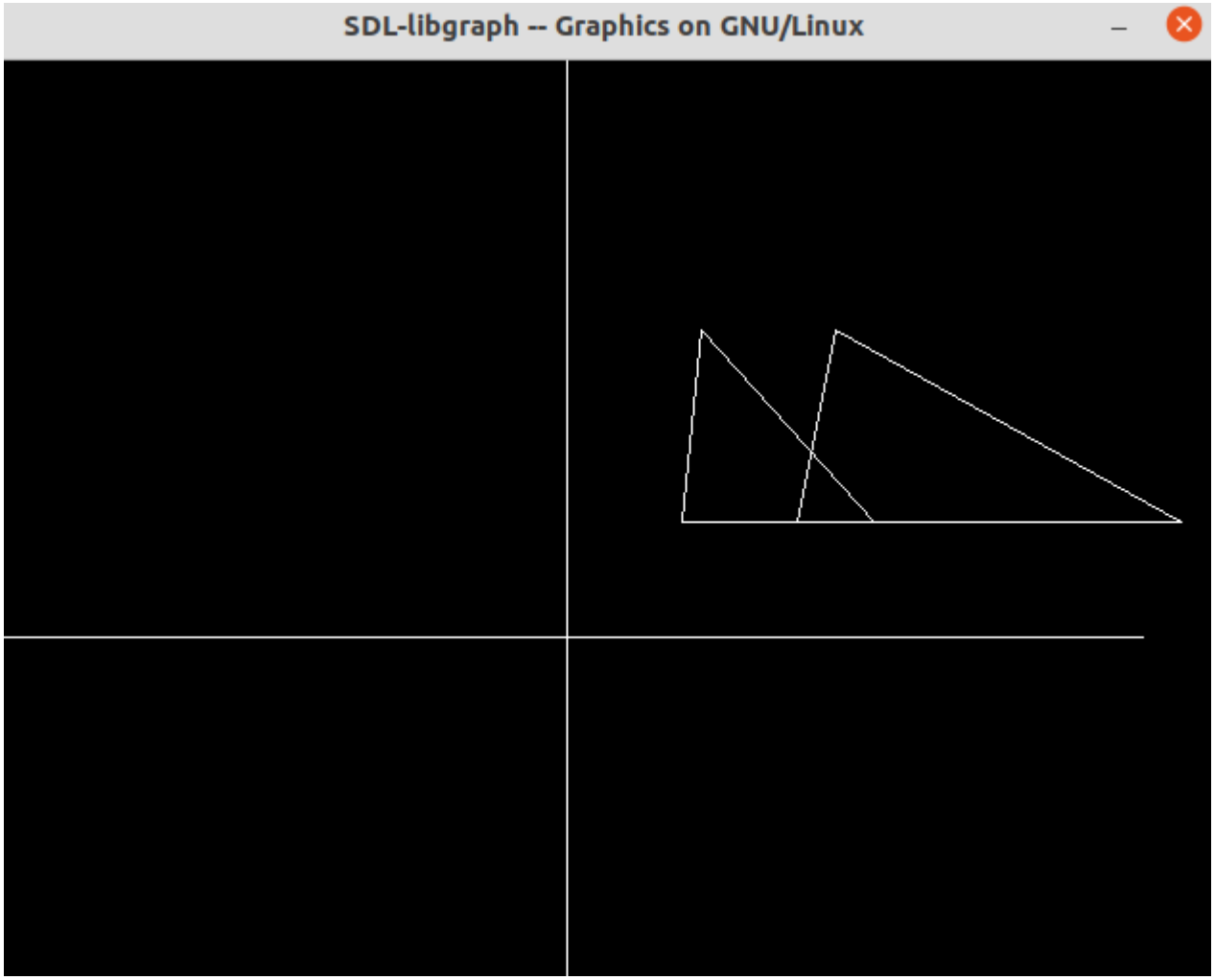
Enter The Coordinates : 60
60
160
60
70
160

Enter your choice
1.Translation

2.Scaling
3.Rotation 2

SCALING OPERATION
Enter value for sx,sy:2
1

OUTPUT:



INPUT:

3.Rotation:

Enter the Number Of Edges: 3

Enter The Coordinates : 60
60
160
60
70
160

Enter your choice
1.Translation
2.Scaling
3.Rotation 3

ROTATION OPERATION
Enter value for angle: 40

OUTPUT: