

THE GRAPHING CALCULATOR



VIRUJ THAKUR

XII – C

ST. COLUMBA'S
SCHOOL

BOARD ROLL
NUMBER:
14658936

INDEX

SERIAL NUMBER	TITLE	PAGE NUMBER
1.	AIM	3
2.	CERTIFICATE	4
3.	HEADER FILES	5
4.	CLASSES	6
5.	CODE	7 – 21
6.	OUTPUT	22 – 28
7.	ACKNOWLEDGEMENT	29
8.	BIBLIOGRAPHY	30

AIM

**To make a working
Graphing Calculator
with the application of
Object Oriented
Programming and File
Handling.**

CERTIFICATE

This is to certify that Viruj Thakur of class 12 - C has successfully completed a project during the session 2019-20.

Submitted for All India Senior Secondary Certificate Computer Science Examination.

(MRS. RITU NAGPAL)

HEADER FILES

graphics.h

fstream.h

conio.h

process.h

stdio.h

math.h

CLASSES

Classes used: graph

- graph:

- Private Members:

- Integer: s, c, o, midx, midy

- Character: i[50],w[3],z[3];

- Floating point: x,y

- void linear() – Plots graph for Linear Equations.

- void quadratic() – Plots graph for Quadratic Equations.

- void cubic() – Plots graph for Cubic Equations.

- void cir() – Plots graph for Circle.

- void ell() – Plots graph for Ellipse.

- Public Members:

- Graph() – Initializes midx and midy with legal values.

- void plotgraph(int) – Calls all the private member functions.

CODE OF THE PROGRAMM

```
#include<conio.h>
#include<stdio.h>
#include<process.h>
#include<graphics.h>
#include<fstream.h>
#include<math.h>
```

```
fstream f;
class graph
{
    int s,c,o;
    char i[50],w[3],z[3];
    float x,y;
    int midx,midy;
    void linear();
    void quadratic();
    void cubic();
    void cir();
    void ell();
    public:
    graph();
    void plotgraph(int);
};
```

```
graph :: graph()
{
    midx=getmaxx()/2;
    midy=getmaxy()/2;
}
```



```

void graph :: linear()
{
    f.open("graph.dat",ios::binary | ios::app);
    x=-midx;
    outtextxy(400,350,"y = mx + c");
    outtextxy(400,365,"Enter 'm' and 'c'");
    float m,c;
    cin>>m>>c;
    f<<"y = "<<m<<"x + "<<c<<"\n";
    f.close();
    setcolor(RED);
    char i[50];
    float x1=-c/m;
    sprintf(i,"x-intercept = %g",x1);
    outtextxy(350,100,i);
    sprintf(i,"y-intercept = %g",c);
    outtextxy(350,115,i);
    int s=50;
    sprintf(i,"y = %gx + ",m);
    outtextxy(s,midy-15,i);
    s+=textwidth(i);
    sprintf(i,"%g",c);
    outtextxy(s,midy-15,i);
    while(x<=midx)
    {
        y=-(m*x + c);
        line(x+midx, y+midy,x+1+midx,
            -(m*(x+1)+c)+midy);
        x++;
    }
}

```

```

void graph :: quadratic()
{
    f.open("graph.dat",ios::binary | ios::app);
    x=-50;
    outtextxy(400,350,"y = ax2 + bx + c");
    outtextxy(400,365,"Enter 'a', 'b' and 'c'");
    float a,b,c;
    cin>>a>>b>>c;
    f<<"y = "<<a<<"x2 + "<<b<<"x + "<<c<<"\n";
    f.close();
    setcolor(RED);
    if((b*b)-(4*a*c)<0)
        outtextxy(350,100,"No x-intercept");
    else
    {
        float x1=(-b+sqrt((b*b)-(4*a*c)))/(2*a);
        float x2=(-b-sqrt((b*b)-(4*a*c)))/(2*a);
        sprintf(i,"x-intercept = %g",x1);
        outtextxy(350,100,i);
        sprintf(i," %g",x2);
        outtextxy(525,100,i);
    }
    sprintf(i,"y-intercept = %g",c);
    outtextxy(350,115,i);
    s=50;
    sprintf(i,"y = %gx2 + ",a);
    outtextxy(s,midy-15,i);
    s+=textwidth(i);
    sprintf(i,"%gx + ",b);
    outtextxy(s,midy-15,i);
    s+=textwidth(i);
}

```

```

    sprintf(i, "%g", c);
    outtextxy(s, midy-15, i);
    while(x<=50)
    {
        y=-(a*x*x + b*x+c);
        line(x+midx, y+midy, x+1+midx, -
(a*(x+1)*(x+1)+b*(x+1)+c)+midy);
        x++;
    }
}

```

```

void graph :: cubic()
{

```

```

    f.open("graph.dat", ios::binary | ios::app);
    x=-10;
    outtextxy(400, 350, "y = ax^3 + bx^2 + cx + d");
    outtextxy(400, 365, "Enter 'a', 'b', 'c' and 'd'");
    float a, b, c, d;
    cin >> a >> b >> c >> d;
    f << "y = " << a << "x^3 + " << b << "x^2 + " << c << "x + " << d << "\n";
    f.close();
    char i[50];
    setcolor(RED);
    float x1, x2, x3, y1, f, g, h, D;
    f = ((3*c)/a - (b*b)/(a*a))/3;
    g = ((2*b*b*b)/(a*a*a) - (9*b*c)/(a*a) + (27*d)/a)/27;
    h = (g*g)/4 + (f*f*f)/27;
    D = 18*a*b*c*d - 4*b*b*b*d + b*b*c*c - 4*a*c*c*c - 27*a*a*d*d;
    if(D>0)
    {
        float i, j, k, l, m, n, p;

```

```

i=sqrt((g*g)/4 - h);
j=pow(i,1/3.);
k=acos(-(g/(2*i)));
l=j*(-1);
m=cos(k/3);
n=sqrt(3)*sin(k/3);
p=(b/(3*a))*(-1);
x1=2*j*cos(k/3) - b/(3*a);
x2=l*(m+n) + p;
x3=l*(m-n) + p;
}
else if(D<0)
{
    float r,s,t,u;
    r=-(g/2) + sqrt(h);
    if(r<0)
    {
        r=fabs(r);
        s=-pow(r,1/3.);
    }
    else
        s=pow(r,1/3.);
    t=-(g/2) - sqrt(h);
    if(t<0)
    {
        t=fabs(t);
        u=-pow(t,1/3.);
    }
    else
        u=pow(t,1/3.);
    x1=s+u - (b/(3*a));
    x2=sqrt(-1);
    x3=sqrt(-1);
}
else
{
    x1=x2=x3=pow(d/a,1/3.)*(-1);
}

```

```

    sprintf(i,"x-intercept = %g",x1);
    outtextxy(350,100,i);
    sprintf(i,"x-intercept = %g",x2);
    outtextxy(350,115,i);
    sprintf(i,"x-intercept = %g",x3);
    outtextxy(350,130,i);
    y1=d;
    sprintf(i,"y-intercept = %g",y1);
    outtextxy(350,145,i);
    s=50;
    sprintf(i,"y = %gx^3 + ",a);
    outtextxy(s,midy-15,i);
    s+=textwidth(i);
    sprintf(i,"%gx^2 + ",b);
    outtextxy(s,midy-15,i);
    s+=textwidth(i);
    sprintf(i,"%gx + ",c);
    outtextxy(s,midy-15,i);
    s+=textwidth(i);
    sprintf(i,"%g",d);
    outtextxy(s,midy-15,i);
    while(x<=10)
    {
        y=-(a*x*x*x + b*x*x + c*x + d);
        line(x+midx,y+midy, x+1+midx,-
(a*(x+1)*(x+1)* (x+1) + b*(x+1)*(x+1) + c*(x+1)
d)+midy);
        x++;
    }
}
void graph :: cir()
{
    f.open("graph.dat",ios::binary | ios::ap
    outtextxy(400,350,"(x - a)^2 + (y - b)^2 = r^2");
    p);
}

```

```

outtextxy(400,365,"Enter 'a','b' and 'r'");
float a,b,r;
cin>>a>>b>>r;
f<<"(x - "<<a<<")^2 + (y - "<<b<<")^2 = "<<r<<"^2\n";
f.close();
char i[50];
setcolor(RED);
float x1,x2,y1,y2;
if(fabs(b)-r>0)
    outtextxy(350,100,"No x-            intercept");
else
{
    x1=sqrt((r*r)-(b*b))+a;
    x2=-sqrt((r*r)-(b*b))+a;
    sprintf(i,"x-intercept =            %g",x1);
    outtextxy(350,100,i);
    sprintf(i,"%g",x2);
    outtextxy(525,100,i);
}
if(fabs(a)-r>0)
    outtextxy(350,115,"No y-            intercept");
else
{
    y1=sqrt((r*r)-(a*a))+b;
    y2=-sqrt((r*r)-(a*a))+b;
    sprintf(i,"y-intercept =            %g",y1);
    outtextxy(350,115,i);
    sprintf(i,"%g",y2);
    outtextxy(525,115,i);
}

```

```

s=30;
sprintf(i,"(x - %g)2 + (y - ",a);
outtextxy(s,midy-15,i);
s+=textwidth(i);
sprintf(i,"%g)2 = ",b);
outtextxy(s,midy-15,i);
s+=textwidth(i);
sprintf(i,"%g2",r);
outtextxy(s,midy-15,i);
circle(a+midx,-b+midy,r);
}

void graph :: ell()
{
    f.open("graph.dat",ios::binary | ios::ap
    outtextxy(350,350,"[(x - x1)/a]2 + [(y
    outtextxy(350,365,"Enter 'x1', 'y1' and
    float x1,a,y1,b;
    cin>>x1>>y1>>a>>b;
    f<<"[(x - "<<x1<<)/"<<a<<"]2 + [(y -
    "<<y1<<)/"<<b<<"]2 = 1\n";
    f.close();
    char i[50];
    setcolor(RED);
    float x2,x3,y2,y3;
    if(fabs(y1)-b>0)
        outtextxy(350,100,"No x-
    else
    {
        x2=sqrt(1-((y1*y1)/(b*b)))*a +
        x3=-sqrt(1-((y1*y1)/(b*b)))*a +
    p);
    - y1)/b]2 = 1");
    'a', 'b'");
    intercept");
    x1;
    x1;

```

```

        sprintf(i,"x-intercept = %g",x2);
        outtextxy(350,100,i);
        sprintf(i,"%g",x3);
        outtextxy(525,100,i);
    }
    if(fabs(x1)-a>0)
        outtextxy(350,115,"No y- intercept");
    else
    {
        y2=sqrt(1-((x1*x1)/(a*a)))*b + y1;
        y3=-sqrt(1-((x1*x1)/(a*a)))*b + y1;
        sprintf(i,"y-intercept = %g",y2);
        outtextxy(350,115,i);
        sprintf(i,"%g",y3);
        outtextxy(525,115,i);
    }
    s=5;
    sprintf(i,"[(x - %g)/",x1);
    outtextxy(s,midy-15,i);
    s+=textwidth(i);
    sprintf(i,"%g]2 + [(y - ",a);
    outtextxy(s,midy-15,i);
    s+=textwidth(i);
    sprintf(i,"%g)/",y1);
    outtextxy(s,midy-15,i);
    s+=textwidth(i);
    sprintf(i,"%g]2 = 1",b);
    outtextxy(s,midy-15,i);
    ellipse(x1+midx,-y1+midy,0,360,a,b);
}

```



```

void graph :: plotgraph(int c)
{
    if(c==1)
        linear();
    else if(c==2)
        quadratic();
    else if(c==3)
        cubic();
    else if(c==4)
        cir();
    else if (c==5)
        ell();
}

```

```

void main()
{
    int gd=DETECT,gm,err;
    initgraph(&gd,&gm,"C:\\TURBOC3\\ BGI");
    err=graphresult();
    if(err!=0)
    {
        cout<<grapherrormsg(err)    <<endl;
        exit(0);
    }
    setcolor(DARKGRAY);
    setbkcolor(WHITE);
    cleardevice();
    graph g;
    int ch;
    char c[50];
    do
    {
        cleardevice();
        settxtstyle(4,0,4);
        outtextxy(65,1,"THE
CALCULATOR");

```

GRAPHING

```

settextstyle(0,0,1);
outtextxy(65,100,"1.
outtextxy(65,200,"2. Quadratic
outtextxy(65,300,"3. Cubic
outtextxy(65,400,"4. Circle");
outtextxy(425,100,"5. Ellipse");
outtextxy(425,200,"6. View
outtextxy(425,300,"7. Clear
outtextxy(425,400,"8. Exit");
cin>>ch;
int i;
char a[50];
cleardevice();
for(i=44;i<=getmaxx();i+=50)
{
    line(i,getmaxy()/2-
1,i,getmaxy()/2+1);
    sprintf(a,"%d",i-
outtextxy(i-
10,getmaxy()/2+5,a);
}
settextjustify(RIGHT_TEXT,
TOP_TEXT);
for(i=14;i<=getmaxy();i+=50)
{
    line(getmaxx()/2-1,
i,getmaxx()/2+1,i);
    sprintf(a,"%d",
getmaxy()/2-i);
outtextxy(getmaxx()/2-
3,i-3,a);

```

Linear Equation");
Equation");
Equation");
saved equations");
Data");
getmaxx()/2);

```
}
```

```
settextjustify(LEFT_TEXT,TOP_TEXT);  
line(1,getmaxy()/2,getmaxx(),getmaxy()/2);  
line(getmaxx()/2,1,getmaxx()/2,getmaxy());  
switch(ch)  
{  
case 1:g.plotgraph(ch);  
    getch();  
    break;  
case 2:g.plotgraph(ch);  
    getch();  
    break;  
case 3:g.plotgraph(ch);  
    getch();  
    break;  
case 4:g.plotgraph(ch);  
    getch();  
    break;  
case 5:g.plotgraph(ch);  
    getch();  
    break;  
case 6:cleardevice();  
    f.open("graph.dat",ios::binary | ios::in);  
    if(f)  
    {  
        f.seekg(0);  
        int n=1;  
        while(!f.eof())  
        {  
            f.getline(c,50,'\n');  
            outtextxy(1,15*n,c);  
            n++;  
        }  
    }  
}
```

```

        getch();
        f.close();
    }
    else
    {
        settextstyle(0,0,2);
        outtextxy(190,100,"NO
EQUATIONS");
        getch();
        break;
    }
    break;
case 7:cleardevice();
    f.open("graph.dat",ios::binary | ios::in);
    if(f)
    {
        remove("graph.dat");
        settextstyle(0,0,2);
        outtextxy(225,100,"DATA
CLEARED");
        getch();
    }
    else
    {
        settextstyle(0,0,2);
        outtextxy(200,100,"NO SAVED
DATA");
        getch();
        break;
    }
    f.close();
    break;
case 8:cleardevice();
    settextstyle(4,0,4);
    outtextxy(65,1,"THE
CALCULATOR");

```

SAVED

ios::in);

CLEARED");

DATA");

GRAPHING

```
default:cleardevice();
    settextstyle(0,0,2);
outtextxy(85,200,"SELECT
CHOICES");
    getch();
}
cleardevice();
setcolor(DARKGRAY);
if(ch>=1 && ch<6)
{
    outtextxy(200,200,"Your
    saved.");
    getch();
}
}while(ch!=8);
closegraph();
}
```

has been

FROM THE GIVEN

Equation

OUTPUT

THE GRAPHING CALCULATOR

1. Linear Equation

5. Ellipse

2. Quadratic Equation

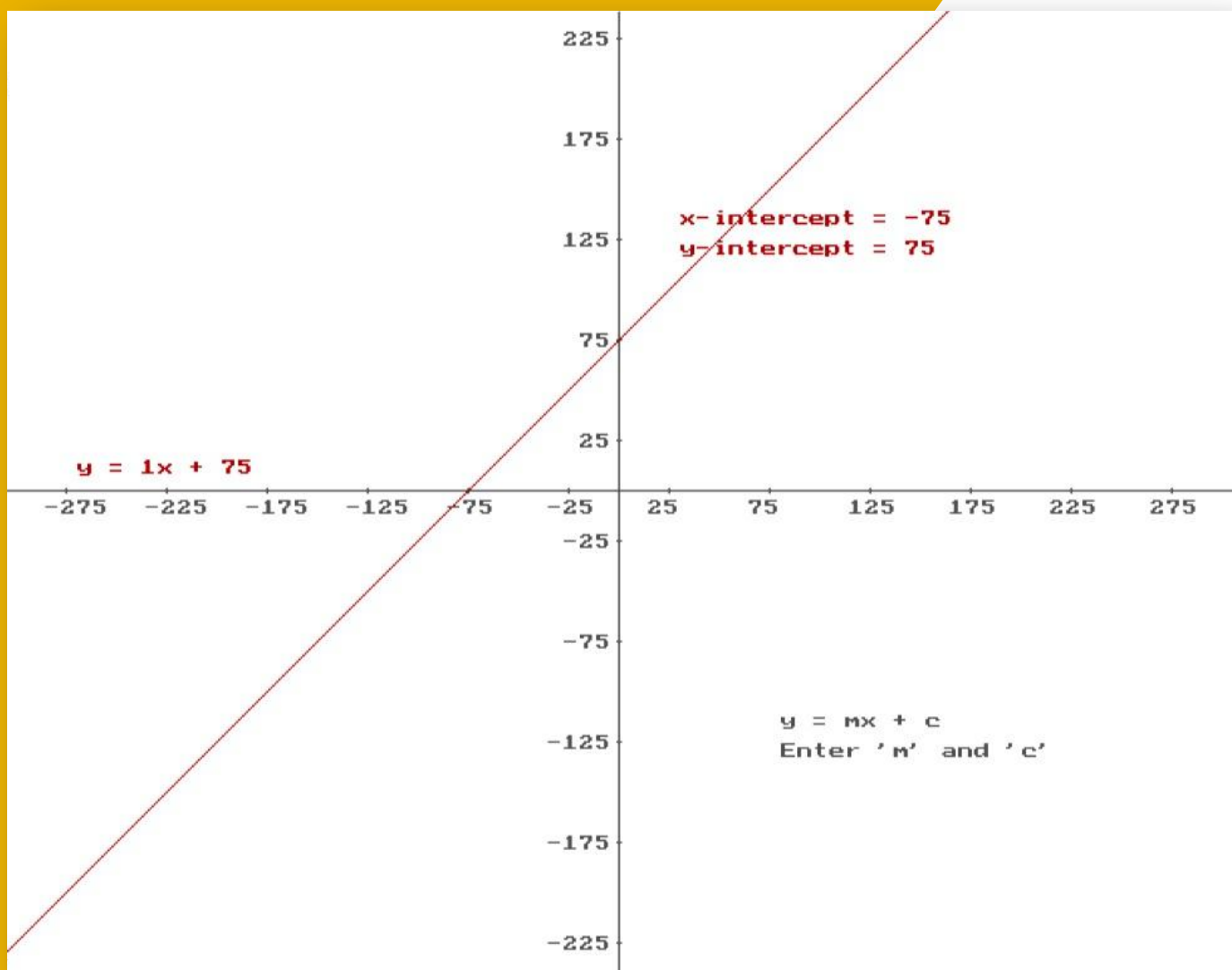
6. View saved equations

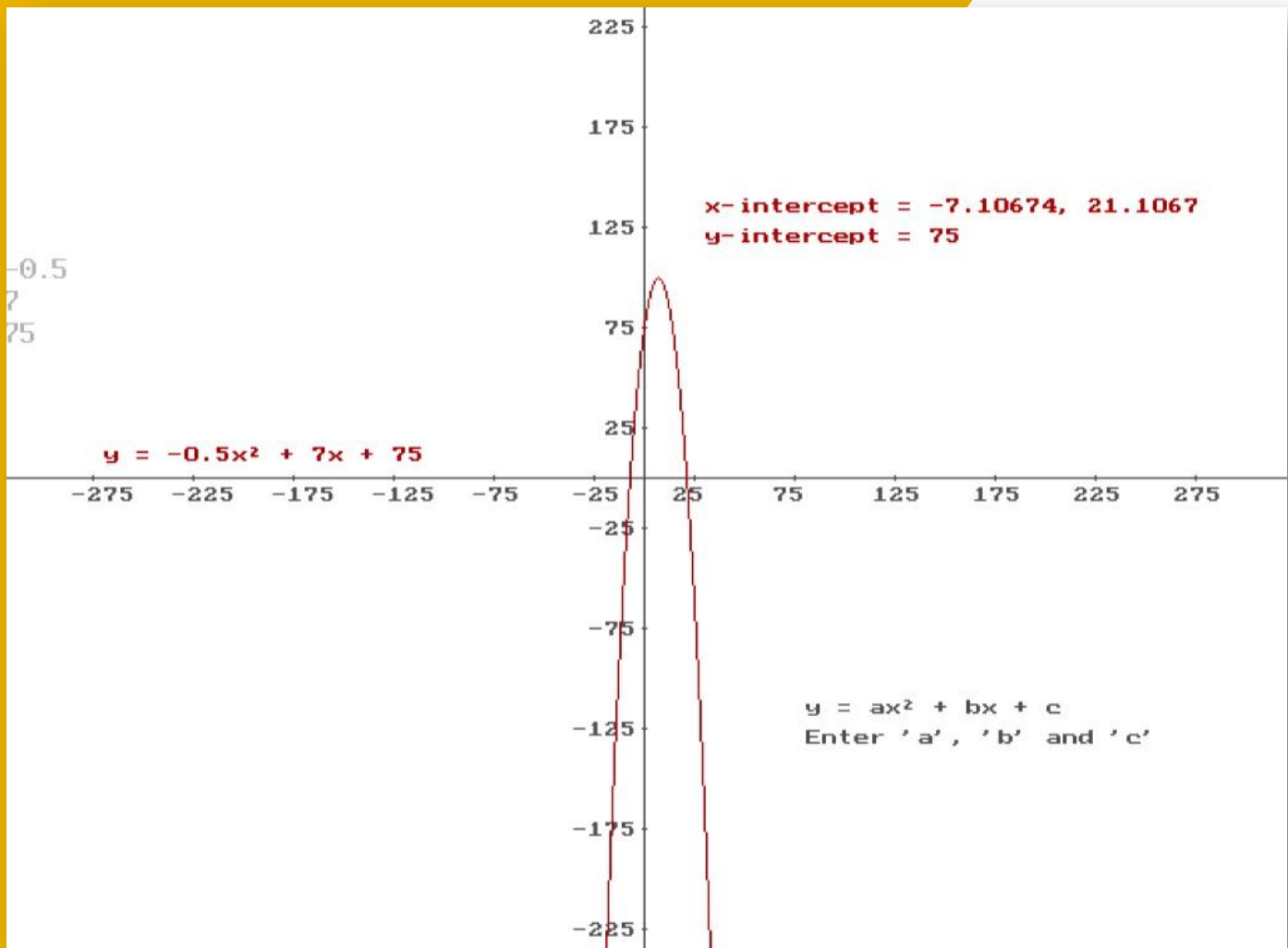
3. Cubic Equation

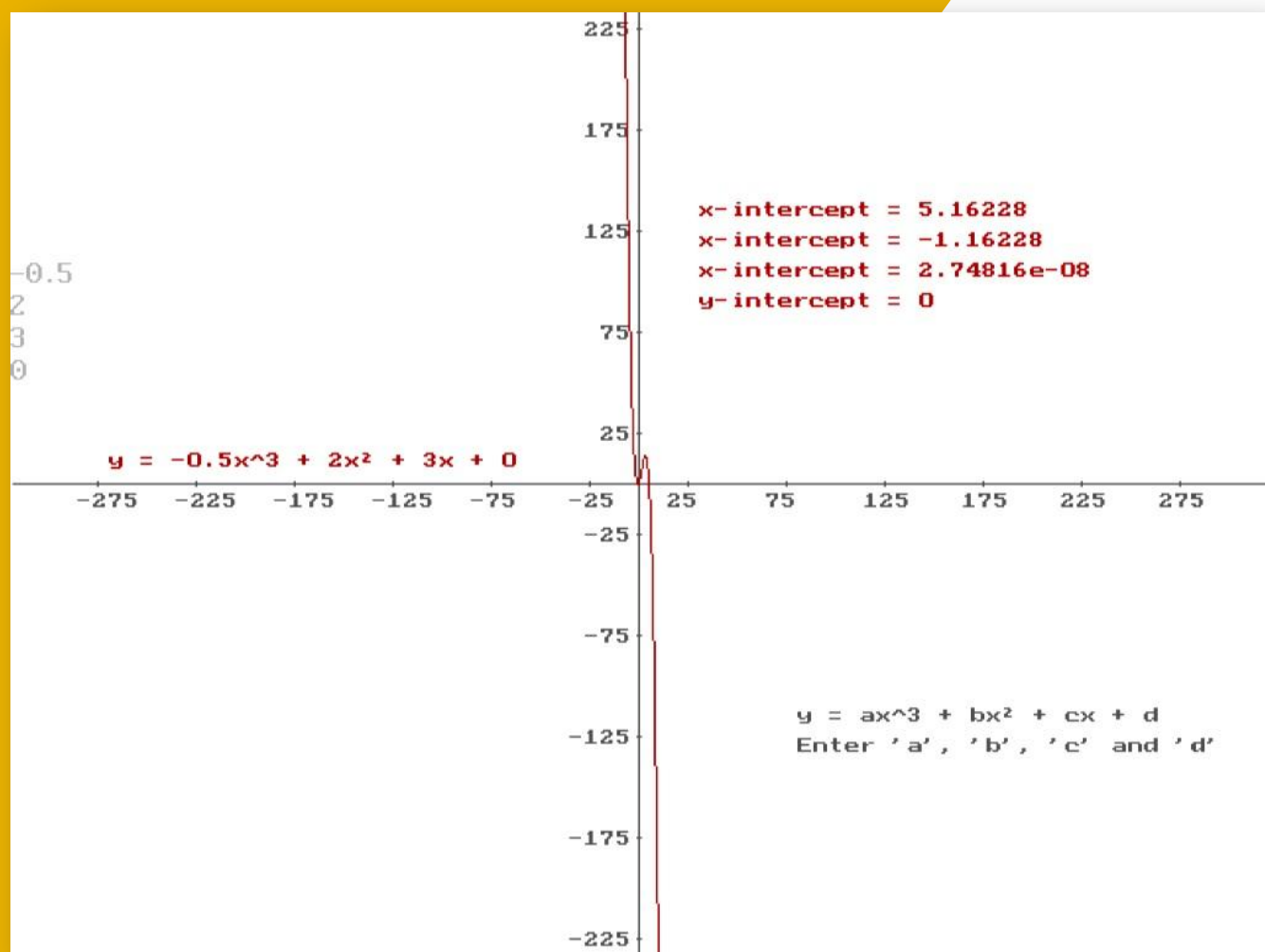
7. Clear Data

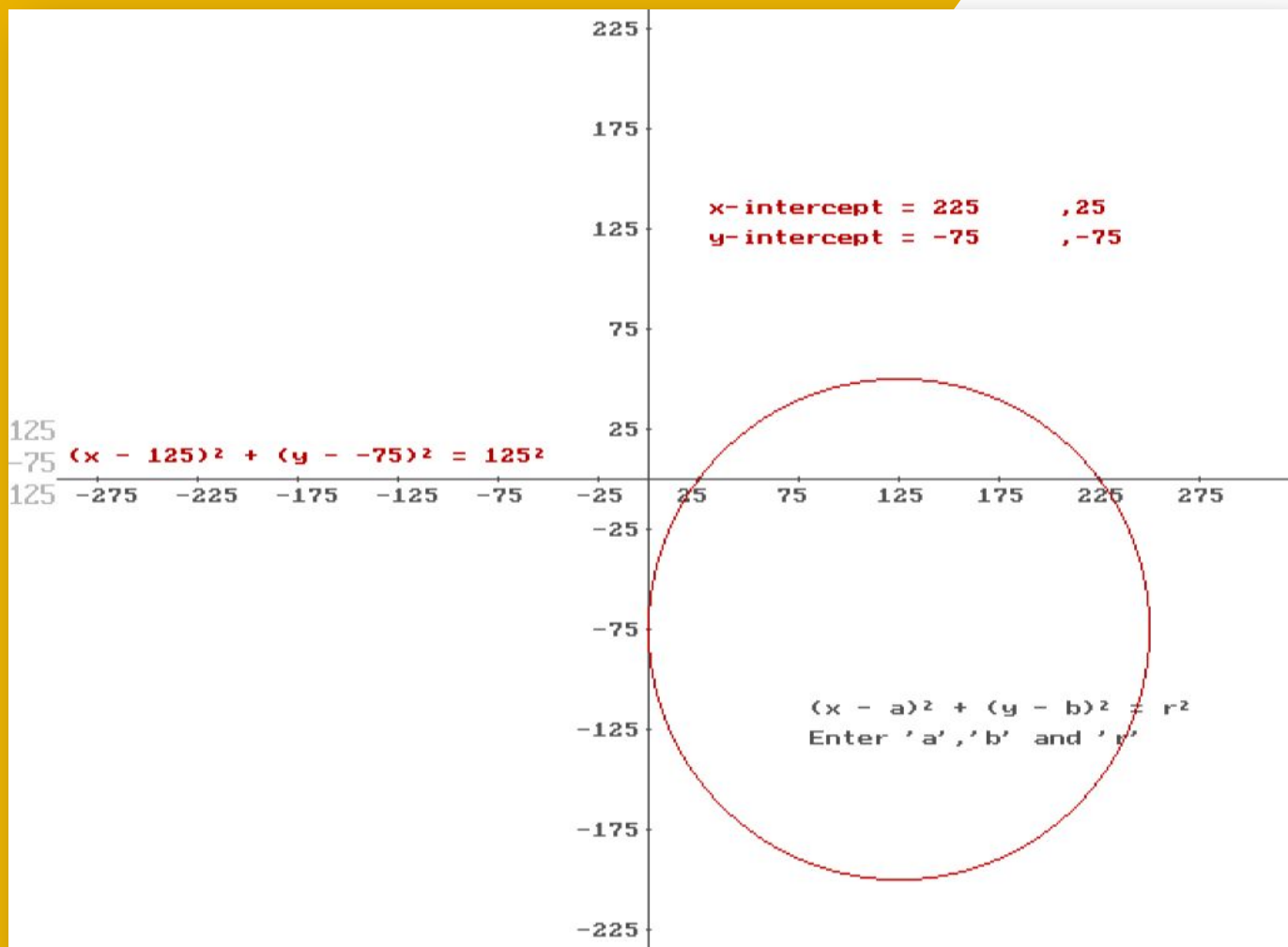
4. Circle

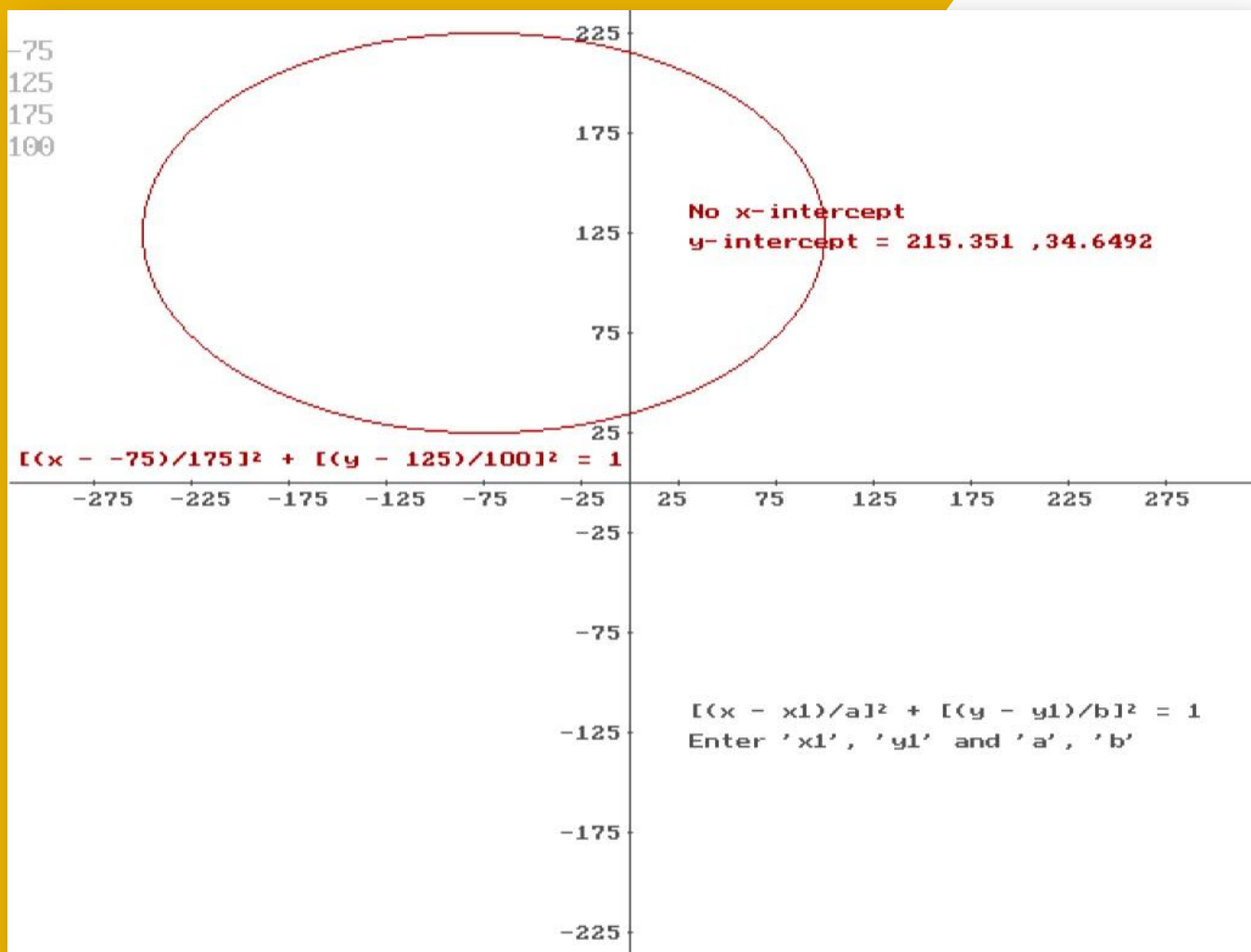
8. Exit











```
y = 75x + 6
y = 0.5x + 2
y = 9x + 5
y = 1x + 75
y = -1x^2 + 3x + 125
y = -0.5x^2 + 7x + 75
y = -0.5x^3 + 2x^2 + 3x + 75
y = -0.5x^3 + 2x^2 + 3x + 0
(x - 125)^2 + (y - -75)^2 = 125^2
[(x - -25)/175]^2 + [(y - 75)/25]^2 = 1
[(x - 25)/125]^2 + [(y - -75)/100]^2 = 1
[(x - -125)/100]^2 + [(y - 75)/175]^2 = 1
[(x - -75)/175]^2 + [(y - 125)/11]^2 = 1
[(x - -75)/175]^2 + [(y - 125)/100]^2 = 1
```

ACKNOWLEDGEMENT

I would like to express my sincere thanks and deep sense of gratitude to my respected teacher Mrs. Ritu Nagpal for her valuable guidance and constant encouragement for the fulfillment of this project.

BIBLIOGRAPHY

○Books

- Sumita Arora. Computer Science with C++. Dhanpat Rai & Co. (P) Ltd., Delhi, 2017.

○Youtube Channels

- Easytuts4you
- VCR Games

○Internet Sites

- <http://www.quora.com>
- <http://www.programiz.com>
- <http://www.cplusplus.com>
- <http://www.1728.org>
- <http://www.wikipedia.com>