//Conics

#include<iostream>

#include<math.h>

using namespace std;

//Detection :

class Detect

{

int p,r;

public:

void setXpower(int p)

{

this -> p=p;

}

int getXpower()

{

return p;

}

void setYpower(int r)

{

this -> r=r;

}

int getYpower()

{

return r;

}

};

//detection ends

//Ellipse

class Ellipse

{

public :

int a,b,i,j ;

Ellipse (float a, float b, float i, float j)

{

this ->a=a;

this ->b =b;

this -> i = i;

this->j =j;

}

virtual void center()

{

}

virtual void focusDistance()

{

}

virtual void vertex()

{

}

};

class GreaterA : public Ellipse

{

public:

GreaterA(float a, float b, float i, float j):Ellipse ( a, b, i, j)

{

}

void center()

{

cout << "the center is at : ("<<i<<","<<j<<")"<<endl;

}

void focusDistance()

{

cout<<"distance of the focus from the center "<< sqrt(pow(a,2)-pow(b,2))<<endl;

}

void vertex()

{

cout << "the vertices are : (+ "<<a+i<<","<<0+j<<")"<<"and (-"<<a+i<<","<<0+j<<")"<<endl;

}

};

class GreaterB : public Ellipse

{

public:

GreaterB(float a, float b, float i, float j):Ellipse ( a, b, i, j)

{

}

void center()

{

cout << "the center is at : ("<<i<<","<<j<<")"<<endl;

}

void focusDistance()

{

cout<<"distance of the focus from the center "<< sqrt(pow(b,2)-pow(a,2))<<endl;

}

void vertex()

{

cout << "the vertices are : ( "<<0+i<<",+"<<b+j<<")"<<"and ("<<0+i<<",-"<<b+j<<")"<<endl;

}

};

//Hyperbola

class Hyperbola

{

public :

int a,b,i,j ;

Hyperbola (float a, float b, float i, float j)

{

this ->a=a;

this ->b =b;

this -> i = i;

this->j =j;

}

virtual void center()

{

}

virtual void focusDistance()

{

}

virtual void vertex()

{

}

virtual void focus()

{

}

};

class GreaterA1 : public Hyperbola

{

public:

GreaterA1(float a, float b, float i, float j):Hyperbola ( a, b, i, j)

{

}

void center()

{

cout << "the center is at : ("<<i<<","<<j<<")"<<endl;

}

void focusDistance()

{

cout<<"distance of the focus from the center "<< sqrt(pow(a,2)+pow(b,2))<<endl;

}

void vertex()

{

cout << "the vertices are : (+ "<<a+i<<","<<0+j<<")"<<"and (-"<<a+i<<","<<0+j<<")"<<endl;

}

void focus()

{

cout << "the focus is at :(+"<<i+sqrt(pow(a,2)+pow(b,2))<<","<<j<<") and ( -"<<i+sqrt(pow(a,2)+pow(b,2))<<","<<j<<")"<<endl;

}

};

class GreaterB1 : public Hyperbola

{

public:

GreaterB1 (float a, float b, float i, float j):Hyperbola( a, b, i, j)

{

}

void center()

{

cout << "the center is at : ("<<i<<","<<j<<")"<<endl;

}

void focusDistance()

{

cout<<"distance of the focus from the center "<< sqrt(pow(b,2)+pow(a,2))<<endl;

}

void vertex()

{

cout << "the vertices are : ( "<<0+i<<",+"<<b+j<<")"<<"and ("<<0+i<<",-"<<b+j<<")"<<endl;

}

void focus()

{

cout << "the focus is at (+"<<i<<","<<j+sqrt(pow(b,2)+pow(a,2))<<") and (-"<<i<<","<<j+sqrt(pow(b,2)+pow(a,2))<<")"<<endl;

}

};

//parabola

class Parabola

{

public :

float a,b,p,r,i,j;

Parabola(float a, float b,float p,float r,float i,float j)

{

this->a =a;

this->b=b;

this->p=p;

this->r=r;

this->i=i;

this->j=j;

}

virtual void vertex()

{

}

virtual void focusDistance ()

{

}

virtual void focus()

{

}

};

class X2 : public Parabola

{

public:

X2(float a, float b,float p,float r,float i,float j):Parabola(a,b,p,r,i,j)

{

}

void vertex()

{

cout<<"the vertex is at ("<<i<<","<<j<<")"<<endl;

}

void focusDistance ()

{

cout<<"the distance of the focus from the vertex is ="<<b/(a\*4)<<endl;

}

void focus()

{

cout<< "the focus is at ("<<i<<","<<j+b/(a\*4)<<")"<<endl;

}

};

class Y2 : public Parabola

{

public:

Y2(float a, float b,float p,float r,float i,float j):Parabola(a,b,p,r,i,j)

{

}

void vertex()

{

cout<<"the vertex is at ("<<i<<","<<j<<")"<<endl;

}

void focusDistance ()

{

cout<<"the distance of the focus from the vertex is ="<<a/(b\*4)<<endl;

}

void focus()

{

cout<< "the focus is at ("<<i+a/(b\*4)<<","<<j<<")"<<endl;

}

};

int main()

{

int n;

cout << "Whaich types of conics problem do you have?"<<endl;

cout<< "1. ellipse "<<endl;

cout<<"2.parabola"<<endl;

cout<< "3. hyperbola "<<endl ;

cout<<"4. Can't detect.need help!!"<<endl;

cin>>n;

//ellipse:

if(n==1)

{

float a,b,i,j;

cout << "enter the value of 'a'"<<endl ;

cin>>a;

cout<< "enter the value of 'b' "<<endl;

cin>> b;

cout<<"enter the number substracted from X"<<endl;

cin>>i;

cout<<"enter the number substracted from Y"<<endl;

cin>>j;

if(a>b)

{

Ellipse \*e;

GreaterA A(a,b,i,j);

e = &A;

e ->center();

e ->vertex();

e ->focusDistance();

}

if(b>a)

{

Ellipse \*e;

GreaterB B(a,b,i,j);

e = &B;

e ->center();

e ->vertex();

e ->focusDistance();

}

}

//hyperbola

else if(n==3)

{

float a,b,i,j;

cout << "enter the value of 'a'"<<endl ;

cin>>a;

cout<< "enter the value of 'b' "<<endl;

cin>> b;

cout<<"enter the number substracted from X"<<endl;

cin>>i;

cout<<"enter the number substracted from Y"<<endl;

cin>>j;

int p;

cout<<"which one is negative? "<<endl;

cout <<"1.Y"<<endl;

cout<<"2.X"<<endl;

cin>>p;

if(p==1)

{

Hyperbola \*h;

GreaterA1 A(a,b,i,j);

h = &A;

h ->center();

h ->vertex();

h ->focusDistance();

h-> focus();

}

if(p==2)

{

Hyperbola \*h;

GreaterB1 B(a,b,i,j);

h = &B;

h ->center();

h ->vertex();

h ->focusDistance();

h->focus();

}

}

//parabola :

else if(n==2)

{

float a,b,p,r,i,j;

cout<<"enter the power of X"<<endl;

cin>>p;

cout<<"enter the power of Y"<<endl;

cin>>r;

cout<<"enter the coefficient of X"<<endl;

cin>>a;

cout<<"enter the coefficient of Y"<<endl;

cin>>b;

cout<<"enter the value substracted from X"<<endl;

cin>>i;

cout<<"enter the value substracted from Y"<<endl;

cin>>j;

if(p==2&& r==1)

{

Parabola \*k;

X2 x(a,b,p,r,i,j);

k =&x;

k -> vertex();

k-> focusDistance ();

k-> focus();

}

else if(r==2 && p==1)

{

Parabola \*k;

Y2 y(a,b,p,r,i,j);

k =&y;

k -> vertex();

k-> focusDistance ();

k->focus();

}

else

{

cout<<"wrong input"<<endl;

}

}

//detection

else if(n==4)

{

float p,r;

cout<<"enter the power of X :"<<endl;

cin>>p;

cout<<"enter the power of Y :"<<endl;

cin>>r;

Detect power;

power.setXpower(p);

power.setYpower(r);

//hyperbola or ellipse detection?

if( power.getXpower()==2 && power.getYpower()==2)

{

int sign;

cout<<"are the sign of X and sign of Y same?"<<endl;

cout<<"1.Yes"<<endl;

cout<<"2.No"<<endl;

cin>>sign ;

//hyperbola detection

if(sign ==2)

{

cout<< " it's a hyperbola"<<endl ;

}

//ellipse or circle?

else if(sign ==1)

{

float coefficient;

cout<< "are the coefficients of X & Y same?"<<endl;

cout<<"1.Yes"<<endl;

cout<<"2.No"<<endl;

cin>> coefficient;

//ellipse

if(coefficient==2)

{

cout<<"it is an ellipse"<<endl;

}

if(coefficient ==1)

{

cout<<"most probably it is a circle.Sorry our program can't help you"<<endl;

}

}

}

//parabola detect

else if((power.getXpower()==1 && power.getYpower()==2) || (power.getXpower()==2&&power.getYpower()==1))

{

cout<<"it is a parabola"<<endl;

}

else

{

cout<<"wrong input"<<endl;

}

//detection ends

}

return 0;

}