```cpp
#include<bits/stdc++.h>

using namespace std;

struct Point{
    double x,y;
    Point(double x=0,double y=0):x(x),y(y){}
};

typedef Point Vector;

const double eps = 1e-10;

Vector operator + (Vector A,Vector B){ return Vector(A.x+B.x,A.y+B.y); }
Vector operator - (Point A,Point B){ return Vector(A.x-B.x,A.y-B.y); }
Vector operator * (Vector A,double p){ return Vector(A.x*p,A.y*p); }
Vector operator / (Vector A,double p){ return Vector(A.x/p,A.y/p); }

bool operator < (const Point &a,const Point &b){
    return a.x < b.x || (a.x == b.x && a.y < b.y);
}

int dcmp(double x){
    if( fabs(x) < eps )  return 0;
    else return x < 0 ? -1 : 1;
}

bool operator == (const Point &a,const Point &b){
    return dcmp(a.x-b.x) == 0 && dcmp(a.y-b.y) == 0;
}

double Dot(Vector A,Vector B){ return A.x*B.x + A.y*B.y; }

double Cross(Vector A,Vector B){ return A.x*B.y - A.y*B.x; }

bool OnSegment(Point p,Point a1,Point a2){
    return dcmp(Cross(a1-p,a2-p)) == 0 && dcmp(Dot(a1-p,a2-p)) < 0;
}

typedef vector<Point> Polygon;

int n = 3,tol;

int isPointInpolygon(Point p,Polygon poly){
    int wn = 0;
    int n = poly.size();
    for(int i=0;i<n;i++)
    {
        if( p == poly[i] )  return 4;   //顶点上
        if( OnSegment(p,poly[i],poly[(i+1)%n]) )  return 3;     //边上
        int k = dcmp(Cross(poly[(i+1)%n] - poly[i],p-poly[i]));
        int d1 = dcmp(poly[i].y - p.y);
        int d2 = dcmp(poly[(i+1)%n].y - p.y);
        if( k > 0 && d1 <= 0 && d2 > 0 )  wn++;
        if( k < 0 && d2 <= 0 && d1 > 0 )  wn--;
```

```
    }
    if( wn != 0 )  return 1;  // 在多边形内
    return 2;   //多边形外
}

void read(int &p){
    int t=0;
    char c;
    c=getchar();
    while( c < '0' || c > '9' )
        c=getchar();
    while( c >= '0' && c <= '9' )
    {
        t=t*10+c-'0';
        c=getchar();
    }
    p = t;
}

int main(int argc,char ** argv){
    int x,y;
    Polygon poly;
    read(x),read(y);
    poly.push_back(Point(x,y));
    read(x),read(y);
    poly.push_back(Point(x,y));
    read(x),read(y);
    poly.push_back(Point(x,y));
    read(x),read(y);
    Point PP(x,y);
    cout<<isPointInpolygon(PP,poly)<<endl;
    return 0;
}

```