

```

1  #include<iostream>
2  using namespace std;
3  struct Point {int x,y;};
4
5  ostream& operator<<(ostream&,Point);
6  istream& operator>>(istream&,Point&);
7
8  Point operator+(Point,Point);
9  Point operator-(Point,Point);
10 int FindOctant(Point);
11 Point ProperPoint(Point,int);
12 void BresenhamLine(Point,Point);
13
14 template <typename DataType>
15 void Swap(DataType&,DataType&);
16
17 int main()
18 {
19     Point InitialPoint,FinalPoint;
20     cin>>InitialPoint>>FinalPoint;
21     BresenhamLine(InitialPoint,FinalPoint);
22     return 0;
23 }
24
25 void BresenhamLine(Point InitialPoint,Point FinalPoint)
26 {
27     Point P=FinalPoint-InitialPoint; //transfer origin
28     int OctantIndex=FindOctant(P);
29     P=ProperPoint(P,OctantIndex); //transform point
30     OctantIndex=(OctantIndex==7?3:(OctantIndex==3)?7:OctantIndex);
31     Point CurrentPoint={0,0};
32     int p=2*P.y-P.x;
33     do
34     {
35         cout<<InitialPoint+ProperPoint(CurrentPoint,OctantIndex)<<
endl;
36         int t=CurrentPoint.y;
37         if(p>0) CurrentPoint.y++;
38         p=p+2*P.y-2*P.x*(CurrentPoint.y-t);
39         CurrentPoint.x++;
40     }
41     while(CurrentPoint.x<=P.x);
42 }
43
44 int FindOctant(Point P)
45 {
46     if(P.x>=0)
47         if(P.y>=0)
48             return (P.y>P.x?2:1);
49         else
50             return (-P.y<P.x?8:7);
51     else
52         if(P.y>0)
53             return (P.y>-P.x?3:4);
54         else
55             return(-P.y>-P.x?6:5);
56 }
57
58
59 Point ProperPoint(Point P,int d)
60 {
61     if(d==2||d==3||d==6||d==7)
62         Swap(P.x,P.y);
63     if(d==7||d==4||d==5||d==6)
64         P.x*=-1;
65     if(d==3||d==8||d==5||d==6)

```

```

66         P.y*=-1;
67     return P;
68 }
69
70 Point operator+(Point P1,Point P2)
71 {
72     Point P;
73     P.x=P1.x+P2.x;
74     P.y=P1.y+P2.y;
75     return P;
76 }
77
78 Point operator-(Point P1,Point P2)
79 {
80     Point P;
81     P.x=P1.x-P2.x;
82     P.y=P1.y-P2.y;
83     return P;
84 }
85
86 ostream& operator<<(ostream& op,Point p)
87 {
88     cout<<" ("<<p.x<<" , "<<p.y<<" ) ";
89     return op;
90 }
91
92 istream& operator>>(istream&ip,Point&p)
93 {
94     cin>>p.x>>p.y;
95     return ip;
96 }
97
98 template <typename DataType>
99 void Swap(DataType& p1,DataType&p2)
100 {
101     DataType p;
102     p=p1;
103     p1=p2;
104     p2=p;
105 }
106
107

```