

```

1 #include <iostream>
2 #include <cmath>
3 using namespace std;
4
5 struct coord_c{
6     double x;
7     double y;
8 };
9
10 struct coord_p{
11     double r;
12     double theta;
13 };
14
15 coord_c converti_in_cartesiana(coord_p P){
16     coord_c Pc;
17
18     Pc.x = P.r * cos(P.theta);
19     Pc.y = P.r * sin(P.theta);
20
21     return Pc;
22 }
23
24 coord_p converti_in_polare(coord_c P){
25     coord_p Pp;
26
27     Pp.r = sqrt(P.x*P.x+P.y*P.y);
28     Pp.theta = atan(P.y/P.x);
29
30     return Pp;
31 }
32
33 int main(){
34     coord_c P1_c,P2_c;
35     coord_p P1_p, P2_p;
36
37     cout << "Inserisci le coordinate cartesiane del primo punto" << endl;
38     cin >> P1_c.x;
39     cin >> P1_c.y;
40
41     P1_p = converti_in_polare(P1_c);
42
43     cout << "Le coordinate polari del primo punto sono ";
44     cout << "(" << P1_p.r << ", " << P1_p.theta << ")" << endl;
45
46     cout << "Inserisci le coordinate polari del secondo punto" << endl;
47     cin >> P2_p.r;
48     cin >> P2_p.theta;
49
50     P2_c = converti_in_cartesiana(P2_p);
51
52     cout << "Le coordinate cartesiane del secondo punto sono ";
53     cout << "(" << P2_c.x << ", " << P2_c.y << ")" << endl;
54
55     return 0;
56 }

```