

```

1
2
3
4  /*
5  *
6  *          OPERACOES COM VETORES
7  *
8  */
9
10
11
12
13
14 double cross(pv a, pv b){//produto vetorial (cross product)
15     // a e b sao vetores
16     return a.x*b.y - a.y*b.x;
17 }
18
19
20
21 bool ccw(pv a, pv b, pv c){//true se a, b, c nessa ordem estao em sentido anti-horario
22     return cross(b-a, c-a) > 0.0;
23 }
24
25
26 bool colinear(pv a, pv b, pv c){//true se a, b, c sao colineares
27     return cross(b-a, c-a) == 0.0;
28 }
29
30
31 double dot(pv a, pv b){//produto escalar (dot product)
32     // a e b sao vetores
33     return a.x*b.x + a.y*b.y;// se a e b sao o mesmo vetor, o resultado vai ser a
34     // norma de a ao quadrado: |a|^2
35 }
36
37 bool multiplos(pv a, pv b){//retorna true false se os vetores forem multiplos
38     // entre si
39     if(a.x == b.x && a.x == 0.0) return true;
40     if(a.y == b.y && a.y == 0.0) return true;
41     return (b.x != 0.0 && b.y != 0.0 && (a.x/b.x == a.y / b.y));
42 }
43
44 pv perp_vec(pv u){//retorna um vetor perpendicular a u
45     return pv(u.y, -u.x);
46 }
47
48
49 double norma(pv u){
50     return sqrt(dot(u, u));
51 }
52
53
54
55 pv unit_vec(pv u){
56     return u*(1.0/norma(u));
57 }

```