

```

    outer = a1*b2 - a2*b1; /*outerは外積*/

    return outer; /*返却値としてouterを返す*/
}

int ccw(struct point p0, struct point p1, struct point p2) {
/*S1, S2から見て、他方の線分の端点それぞれの位置が時計回りかそうでないかを判断する関数*/
    struct segment z1, z2; /*変数の宣言*/
    int op;

    z1.p1 = p0; /*線分z1は、線分p0p1*/
    z1.p2 = p1;
    z2.p1 = p1; /*線分z2は、線分p1p2*/
    z2.p2 = p2;

    op = outer_product(z1, z2); /*線分z1, z2をouter_product関数にわたし、返された値をopに
代入*/

    if( op > 0 ) /*z1とz2の外積が正なら*/
        return 1; /*返却値として1を返す*/
    else if( op < 0 ) /*z1とz2の外積が負なら*/
        return -1; /*返却値として-1を返す*/
    else /*z1とz2の外積がなら*/
        return 0; /*返却値として0を返す*/
}

int seg_i(struct segment s1, struct segment s2) {
/*2線分が交差するかどうか判断する関数*/
    int a, b, c, d; /*変数の宣言*/

    a = ccw(s1.p1, s1.p2, s2.p1); /*S1からみたS2のひとつめの端点の位置*/
    b = ccw(s1.p1, s1.p2, s2.p2); /*S1からみたS2のふたつめの端点の位置*/
    c = ccw(s2.p1, s2.p2, s1.p1); /*S2からみたS1のひとつめの端点の位置*/
    d = ccw(s2.p1, s2.p2, s1.p2); /*S2からみたS1のふたつめの端点の位置*/

    if( a*b==1 || c*d==1 ) { /*a, bがともに-1、もしくはともに1 又はc, dがともに-1、も
しくはともに1*/
        return 0; /*つまり、ひとつの線分に対して他方の線分の両端点の
位置が同じ側ならば0を返す*/
    }
    else
        return 1; /*そうでないなら、1を返す*/
}

```

本当か？

-8