

TEMPLATES 01-08pm**I. Hàm định hướng**

```
int ccw(pt A, pt B, pt C) {
    double t = (B - A).cross(C - A);
    if (t > 0) return 1;
    if (t < 0) return -1;
    return 0;
}
```

II. Hàm tính giao của hai đường thẳng AB và CD:

Gọi M là giao hai đường thẳng AB và CD (biết chắc chắn hai đường thẳng này cắt nhau). Ta có:

$$\overrightarrow{AC} = \overrightarrow{AM} + \overrightarrow{MC} \rightarrow x.\overrightarrow{AB} + y.\overrightarrow{CD} = \overrightarrow{AC} \rightarrow x = \frac{\overrightarrow{AC} \times \overrightarrow{CD}}{\overrightarrow{AB} \times \overrightarrow{CD}}$$

```
pt Giao(pt A, pt B, pt C, pt D) {
    double x = (C - A).cross(D - C) / (B - A).cross(D - C);
    pt M = A + pt{x*(B.x - A.x), x*(B.y - A.y)};
    return M;
}
```

III. Hàm tính giao của đa giác P với nửa đường của đường thẳng MN

```
vector<pt> Cat(pt M, pt N, vector<pt> P) {
    vector<pt> result;
    P.push_back(P[0]);
    int sPre = ccw(P[0], M, N);
    for(int i = 1; i < P.size(); ++i) {
        int sNxt = ccw(P[i], M, N);
        if (sNxt * sPre < 0)
            result.push_back(Giao(M, N, P[i - 1], P[i]));
        if (sNxt >= 0) result.push_back(P[i]);
        sPre = sNxt;
    }
    return result;
}
```

IV. Hàm tính diện tích đa giác P

```
double Area(vector<pt> &P) {
    int n = P.size();
    double res = P[n - 1].cross(P[0]);
    for(int i = 0; i < n - 1; ++i) res += P[i].cross(P[i + 1]);
    return 0.5 * abs(res);
}
```