

Задача В3. Зелени площи

В град Ш. има много зелени площи. Всяка от зелените площи е с формата на изпъкнал многоъгълник. За да се поддържат зелените площи в изряден вид, кметът на града решил да помоли Директора на Математическата гимназия в Ш. да възложи на всеки ученик поддържането на поне една от тях. Директорът пък решил да възложи на ученика от 9 клас с профил Информатика Шибил да направи програма, която по случаен начин да разпредели площите между учениците. Шибил, обаче, решил да се възползва от възможността и да остави за себе си най-малката площ. Понеже току що е започнал да учи програмиране, задачата му се сторила много трудна. Помогнете му, като напишете програма **GREEN**, която да намира зелената площ с минимално лице.

Входните данни ще бъдат зададени на стандартния вход. На всеки ред ще бъде зададено описанието на един от многоъгълниците. То започва с броя M на върховете на многоъгълника ($3 \leq M \leq 100$), последван от M -те двойки целочислени координати на върховете (в правоъгълна координатна система), зададени в посока на часовниковата стрелка. Всички числа са разделени с по един интервал. След описанието на последния многоъгълник следва ред, съдържащ само числото 0.

На стандартния изход програмата трябва да изведе номера на многоъгълника с най-малко лице, като номерацията на многоъгълниците започва от 1 и следва реда, по който те са зададени на входа. Ако два или повече многоъгълника са с едно и също минимално лице, да се изведе номера на този от тях, който има по-малко върхове. Ако два или повече многоъгълника са с едно и също минимално лице и един и същ брой върхове, да се изведе този от тях, който има по-малък номер.

Пример. Вход:

4 13 11 13 21 23 21 23 11

3 -15 -5 0 5 15 -5

4 -5 -5 -5 5 5 5 5 -5

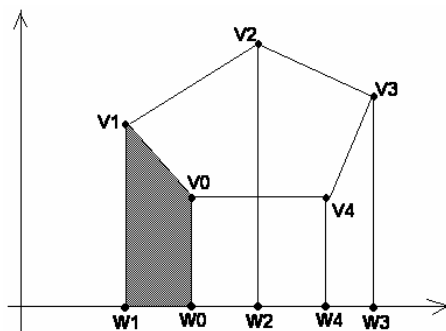
0

Изход:

1

Решение:

Задачата за намиране на лице на многоъгълник е основна за изчислителната (комбинаторната) геометрия. Освен пряко по предназначение, намирането на лицето на многоъгълник може да се окаже важна стъпка при решаване на други интересни задачи. Преди да се спрем на авторското решение, нека да отбележим, че за намирането на лице на изпъкнал многоъгълник могат да се предложат различни алгоритми. Една възможност е да разбием многоъгълника с върхове $V_0, V_1, \dots, V_{N-1}$ на триъгълници $V_0V_1V_2, V_0V_2V_3, \dots, V_0V_{N-2}V_{N-1}$, да намерим лицето на всеки триъгълник, например с използване на формулата на Херон и да пресметнем лицето на изпъкналия многоъгълник като сума от лицата на триъгълниците. Подобен подход има сериозен недостатък – изчисленията при такъв алгоритъм са свързани със загуба на точност. При формулата на Херон ще са ни необходими дължините на страните на триъгълника, при намирането на които ще се наложи използването на коренуване, а и при пресмятането на окончателния резултат ще се наложи да се коренува още веднъж. В изчислителната геометрия коренуването, както делението и пресмятането на функции като $\sin$, $\cos$, $\lg$, $\arctg$ и т.н., често води до загуба на точност. Затова използването на такива пресмятания трябва да се избягва или да се сведе до минимум.



Фиг. 1

За решаване на задачата се предлага алгоритъм, който не предизвиква подобни изчислителни проблеми – т.н. *ориентирани лица*. Същността на алгоритъма е в следната формула:

$$S=|(x_1-x_0)*(y_1+y_0)/2+(x_2-x_1)*(y_2+y_1)/2+\dots+(x_{N-1}-x_{N-2})*(y_{N-1}+y_{N-2})/2+(x_0-x_{N-1})*(y_0+y_{N-1})/2|$$

където (x_i, y_i) , $i=0, 1, \dots, N-1$, са координатите на върховете на изпъкналия многоъгълник, в реда, в който се срещат при обхождането му (по посока на часовниковата стрелка или в обратна посока). На верността на тази формула няма да се спираме тук, а само ще отбележим, че стойността на израза $(x_i-x_{i-1})*(y_i+y_{i-1})/2$ е ориентираното лице на трапеца $V_{i-1}V_iW_iW_{i-1}$, където W_i и W_{i-1} са ортогоналните проекции на V_i и V_{i-1} върху абсцисната ос (вж. Фиг. 1 където е заштриховано лицето на трапеца $V_0V_1W_1W_0$). Ориентирано лице е лицето на трапеца със знак плюс или минус в зависимост от това дали W_i е по-надясно от W_{i-1} върху абсцисната ос. Така ориентираното лице на заштрихования на Фиг.1 трапец е отрицателно и при пресмятането ще бъде извадено от положителното ориентирано лице на трапеца $V_1V_2W_2W_1$.

Забележете още, че въпросната формула е валидна не само за изпъкнали многоъгълници, но и за неизпъкнали несамопресичащи се.

И така, за пресмятане на лицето на един многоъгълник не е необходимо да запомняме координатите на върховете му в паметта. Достатъчно е да помним само координатите на началната точка, за да пресметнем последното събираемо на израза. За намирането на минимума също не е необходимо да запомняме намерените лица.

На Фиг. 2 е показан програмен фрагмент с основните стъпки на алгоритъма.

```
minface=(double)MAX; minv=MAX; minn=MAX;
i=1;
while (1)
{
    cin>>M;
    if (M==0) break;
    face=0.;
    cin>>x1>>y1; x0=x1; y0=y1;
    for (j=2; j<=M; j++)
    {
        cin>>x2>>y2;
        face+=(x2-x1)*(y2+y1)/2;
        x1=x2; y1=y2;
    }
    face+=(x0-x1)*(y0+y1)/2;
    if (face<0) face= -face;
    if ((face<minface) || (face==minface && M<minv))
    {minface=face; minv=M; minn=i++;}
}
cout<<minn<<endl;
```

Фиг. 2