**Задача 1.** *«Штраф за ліві повороти». У місті Х водіям заборонено виконувати ліві повороти. За кожен такий поворот водій має сплатити штраф у вигляді М гривень. Для стеження за водіями у місті встановлена комп'ютерна система, що фіксує координати автомобіля на початку руху, наприкінці руху та під час повороту.*

*Вхідні дані: N* – кількість зафіксованих координат автомобіля, *(xi, yi)* – координати автомобіля у процесі руху, *i*=1,2, …, *N*, де *(x1, y1)* – точка початку руху, *(xN, yN)* – остання точка маршруту автомобіля.

*Потрібно* за заданою послідовністю координат руху обчислити суму штрафу водія.

Траєкторію руху автомобіля можна представити у вигляді ламаної, що складається зі спрямованих відрізків із точок *(xi, yi)* у точки *(xi+1, yi+1), i*=1,2,…,*N*-1. Поворот вважається лівим, якщо напрямок поточного відрізка шляху *ai+1*змінюється щодо попереднього відрізка *ai* у ліву сторону, тобто проти годинникової стрілки.

Таким чином, розв'язання задачі зводиться до обчислення кількості пар ділянок шляху *ai* та *ai+1*, для яких виконується умова [*ai × ai+1*]>0. Координати векторів ai та ai+1 обчислюються через координати точок *(xi, yi): ai = (xi - xi-1, yi - yi-1), ai+1=(xi+1 - xi, yi+1 - yi),* отже, *[ai × ai+1]= (xi - xi-1) (yi+1- yi) – (yi - yi-1)(xi+1 - xi), i=2, …, N-1.*

**Задача 2.** *"Тут буде місто-сад". Мешканці одного будинку міста Х вирішили висадити у себе на подвір'ї кілька дерев. Оскільки мешканці не змогли домовитися, як мають бути розташовані насадження, то кожен посадив дерево в тому місці двору, де йому захотілося. Після проведення насаджень отриманий садок вирішили обгородити парканом. Але доки дошки не привезли, дерева обв'язали однією довгою мотузкою.*

*Вхідна інформація:* *N* – кількість дерев у саду, *(xi, yi)* – координати дерев, *i*=1,2, …, *N*. Оскільки висадили молоді саджанці, їх товщиною можна знехтувати.

*Потрібно* визначити, до яких посаджених дерев треба прив'язати мотузку так, щоб усі дерева опинилися всередині обгородженої зони, а довжина мотузки була мінімальною.

Ця задача зводиться до визначення для заданої множини точок на площині опуклої оболонки, тобто опуклого багатокутника з вершинами у деяких точках із заданої множини, що охоплює всі його точки.

Будуватимемо опуклу оболонку в порядку обходу ділянки за годинниковою стрілкою. Знайдемо найбільш ліву точку М0=*(x0, y0)*, *x0*=min{*xi*}. Якщо таких точок декілька, то візьмемо найнижчу з них. Ця точка належить шуканій опуклій оболонці. Задамо початковий вектор *a0* з початком у точці *(x0, y0),* паралельний осі Oy.

Наступною точкою оболонки буде така точка М1, щоб вектор *a1* з початком у точці М0 і кінцем у точці М1 утворював з початковим вектором *a0* мінімальний кут. Якщо таких точок декілька, то вибирається точка, відстань до якої максимальна.

Далі процес продовжуємо, тобто шукаємо точку М2 з мінімальним кутом між вектором *a1* і вектором *a2* з початком у точці М1 і кінцем у точці М2, потім точку М3 і т.д. Процес припиняємо, коли дійдемо до першої обраної точки або кількість точок в оболонці дорівнюватиме *N*.

Для визначення кута між векторами використовується скалярний добуток. Причому сам кут можна не обчислювати, оскільки мінімальному куту відповідає максимальний косинус кута.

**Задача 3.** *«Заєць». Неподалік міста Х знаходиться зоосад. Місцевий мешканець, заєць, хаотично стрибаючи, залишив слід у вигляді замкнутої ламаної, яка перетинає сама себе й охоплює територію його володіння. Знайти площу мінімального за площею опуклого багатокутника, описаного навколо цієї території.*

У цій задачі необхідно не тільки знайти опуклу оболонку множини точок, але й обчислити площу опуклого багатокутника із заданим набором вершин.

*Вхідні дані: N* – кількість вершин опуклого багатокутника, *(xi, yi)* – координати вершин, *i* = 1,2, ..., *N*.

*Потрібно* визначити площу опуклого N-кутника.

Площа N-кутника може бути обчислена як сума площ трикутників, із яких складено N-кутник. Для знаходження площі трикутника використовуємо векторний добуток. Довжина векторного добутку векторів, як відомо, дорівнює подвоєній площі трикутника, побудованого на цих векторах. Нехай вершини трикутника розташовані в точках A = *(x1, y1),* B = *(x2, y2),* C = *(x3, y3).* Поєднаємо початок координат із першою точкою. Векторний добуток дорівнює

[AB × AC] = $\left|\begin{matrix}i&j&k\\x\_{2}-x\_{1}&y\_{2}-y\_{1}&0\\x\_{3}-x\_{1}&y\_{3}-y\_{1}&0\end{matrix}\right|$,

відповідно, площа трикутника дорівнює

SABC = 1/2*((x2 - x1)(y3 - y2) - (y2-y1)(x3-x2))*.

Значення величини SABC може бути як додатним, так і від’ємним числом, оскільки воно залежить від взаємного розміщення векторів AB та AC, тому кажуть, що площа орієнтовна.

Для знаходження площі *N*-кутника останній потрібно розбити на трикутники та знайти суму орієнтовних площ цих трикутників. Розбиття *N*-кутника на трикутники можна провести так: зафіксувати одну з вершин *N*-кутника, наприклад, першу A1=(x1, y1) і розглядати всі трикутники A1A*i* A*i+1*, *i*=2, 3,…, *N*-1.

Зауважимо, що аналогічним способом можна знаходити площу довільного багатокутника.

Використання властивості орієнтації площі трикутника, обчисленої за векторним добутком, дозволяє визначити, чи є заданий багатокутник опуклим. Для опуклого багатокутника всі трикутники, утворені трійками сусідніх вершин у порядку їх обходу, мають одну орієнтацію. Тому перевірка багатокутника на опуклість може бути проведена за допомогою послідовного порівняння знаків векторних добутків для всіх пар сусідніх сторін багатокутника.

**Задача 4.** *«Тигр у загоні». Неподалік міста Х знаходиться заповідник, у якому мешкають уссурійські тигри. Працівники заповідника дуже переживають, коли тигр залишає зону під охороною. Програма охорони уссурійських тигрів передбачає постачання кожному тигру нашийника з радіомаяком. Сигнал від тигрового радіомаяка надходить до центру охорони та дозволяє визначити розташування тигра. Територія заповідника є довільним багатокутником.*

*Вхідні дані: N* – кількість вершин багатокутника, що задає заповідник, *(xi, yi)* – координати його вершин, *i* = 1,2, …, *N*. *(X, Y)* – координати точки, де знаходиться тигр.

*Потрібно* визначити, чи знаходиться тигр на території заповідника, чи треба терміново споряджати рятувальну експедицію.

Дуже часто при вирішенні задач геометричного змісту потрібно перевірити, чи лежить задана точка всередині або поза багатокутником. Таким способом можна вирішити, наприклад, задачу про Бармаглота, перевіряючи кожну точку Бармаглота відносно ковдри-багатокутника. Є багато способів перевірки належності точки багатокутнику, однак ми наведемо тут один із них, заснований на використанні добутку векторів.

Ідея методу полягає в тому, щоб визначити суму кутів, під якими з точки, що перевіряється, видно сторони багатокутника. Якщо точка лежить усередині багатокутника, то сумарний кут дорівнює 2π (точка Р на малюнку), якщо ж точка лежить поза багатокутником, то сума кутів не дорівнює 2π (точка Q).

Таким чином, для розв’язання треба перебрати у циклі послідовно всі вершини багатокутника і знайти суму кутів між векторами PA*i* та PA*i+1*, *i*=1,2, …, *N*. Не забудьте додати кут між векторами PA*N* та PA1. Для визначення величини кута між векторами нам знадобиться формула скалярного добутку.

Оскільки сторони багатокутника повинні розглядатися послідовно, у порядку обходу вершин, то при знаходженні сумарного кута слід враховувати взаємне розташування векторів. Кут, під яким сторону видно з досліджуваної точки, може бути як додатним, і від’ємним. Для визначення знака кута скористаємося векторним добутком. Знак векторного добутку визначить знак конкретного кута у загальній сумі.