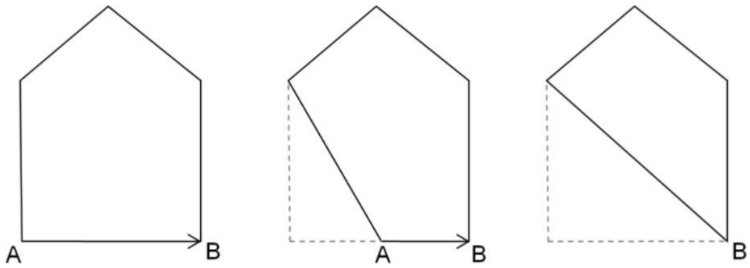
Task: POL
Polygon compression

english

ONTAK 2022, day 2. Available memory: 512 MB. *30.06.2022*

Consider a convex polygon with n vertices (and thus n sides). You are going to perform a sequence of $n-1$ moves that will compress the polygon into a single point.

An allowed move consists of selecting two adjacent vertices A and B, and dragging A onto B. At the end of the move the two vertices merge into one, yielding a polygon with one fewer vertices. The new vertex still has the label B (the label A has disappeared).



The cost of a move is equal to the Euclidean distance between A and B. Given such a polygon, you are asked to solve one of two related problems:

1. Find the smallest total cost of such a sequence of moves.
2. Find one sequence of $n-1$ moves with the smallest total cost.

Input

Each input starts with a line containing a number $p \in \{1,2\}$: the problem you are solving.

The second line of the input contains t – the number of test cases that follow. (In each of them you are solving problem p.)

Each test case starts with a line containing the number n. This is followed by n lines, each containing the coordinates x_i, y_i of one of the polygon's vertices.

All coordinates are integers. All vertices are distinct. The vertices are given in counter-clockwise order.

The polygon is guaranteed to be convex, but not necessarily strictly convex – i.e., collinear vertices are allowed.

In each test case $t \leq 5$, $n \leq 2000$, and all coordinates satisfy $|x_i|, |y_i| \leq 10^6$.

Output

For each test case output its solution.

If solving the problem $p = 1$, the solution is a single line with a single real number: the minimum total cost. (We recommend to output at least ten decimal places.)

If solving the problem $p = 2$, the solution is a sequence of $n-1$ lines, each of the form A_j B_j: for each move one line containing the vertex you want to drag and its neighbor that is the destination.

To avoid rounding errors, in each test case your answer will be considered correct if the minimum total cost in your solution differs from the reference solution by at most 10^{-4}. (If you just calculate all distances as doubles and add them together without any special considerations, the final value you'll obtain should always be comfortably within this range.)

Grading

There are twelve subtasks.

Subtasks 1, 2, 3, 4, 5, 6 are worth 4, 8, 12, 12, 12 and 32 points. They all have $p = 1$ and the maximum value of n in them is 7, 15, 50, 100, 500, and 2000.

For each $i \leq 6$, subtask $6+i$ has $p = 2$, the same maximum n as subtask i, and it is worth one quarter of points of subtask i.

(In total, you get 80 points for solving all tests with $p = 1$ and the remaining 20 for tests with $p = 2$.)

Examples

For the input data:

```
1
2
4
0 0
1 0
1 1
0 1
5
0 0
8 0
8 10
4 20
0 10
```

a correct result is:

```
3.000000000000
36.770329614269
```

For the input data:

```
2
2
4
0 0
1 0
1 1
0 1
5
0 0
8 0
8 10
4 20
0 10
```

a correct result is:

```
3 2
4 1
2 1
4 3
5 3
3 2
2 1
```

Zadanie: POL
Kompresja wielokąta

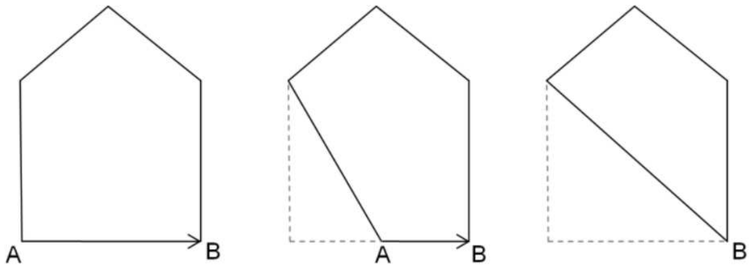
polish

ONTAK 2022, dzień 2. Dostępna pamięć: 512 MB. *30.06.2022*

Rozważmy wielokąt wypukły o n wierzchołkach (i tyluż bokach). Musisz wykonać na nim $n - 1$ operacji, które skompresują wielokąt do pojedynczego punktu.

Dozwoloną operacją jest wybór dwóch sąsiednich wierzchołków A i B, po czym przeciągnięcie punktu A na miejsce punktu B. Po tej operacji wierzchołki łączą się w jeden, przez co wielokąt ma o jeden wierzchołek mniej.

Nowy wierzchołek nazywa się teraz B (czyli A znika).



Koszt operacji jest równy euklidesowej odległości między A i B. Mając taki wielokąt, musisz rozwiązać dwa powiązane ze sobą problemy:

1. Znajdź minimalny koszt ciągu operacji doprowadzającego wielokąt do punktu.
2. Wypisz ciąg takich $n - 1$ operacji doprowadzających do punktu, o minimalnym koszcie.

Wejście

Pierwszy wiersz wejścia zawiera liczbę $p \in \{1, 2\}$: numer problemu, który rozwiązujesz.

Drugi wiersz zawiera t – liczbę zestawów danych, których opisy następują później. (We wszystkich rozwiązujesz ten sam problem p.)

Każdy zestaw rozpoczyna się wierszem zawierającym liczbę wierzchołków n. Potem następuje n wierszy, każdy zawierający parę współrzędnych x_i, y_i jednego z wierzchołków wielokąta. Wszystkie współrzędne są całkowite, żadne dwa wierzchołki nie są w tym samym punkcie. Wierzchołki są podane w kolejności przeciwnie do ruchu wskazówek zegara.

Wielokąt jest wypukły, ale nie musi być ściśle wypukły (czyli dozwolone są trzy współliniowe wierzchołki).

W każdym zestawie danych zachodzi $t \leq 5$, $n \leq 2000$, a wszystkie współrzędne spełniają warunki $|x_i|, |y_i| \leq 10^6$.

Wyjście

Dla każdego zestawu wypisz jego rozwiązanie. Zależnie od p będzie to:

Dla $p = 1$ rozwiązaniem jest jedna liczba rzeczywista – minimalny koszt odpowiedniego ciągu operacji. (Zalecamy wypisanie go do co najmniej dziesiątego miejsca po przecinku).

Dla $p = 2$ rozwiązaniem jest ciąg $n-1$ wierszy, każdy zawierający parę liczb A_j B_j opisujących pojedynczą operację: numer wierzchołka przeciąganego i jego sąsiada będącego celem ruchu.

Aby uniknąć błędów precyzji, w każdym zestawie Twoja odpowiedź zostanie uznana za poprawną, jeśli podany przez Ciebie koszt nie różni się od wzorcowej odpowiedzi o więcej niż 10^{-4}. (Jeśli po prostu obliczysz wszystkie odległości w zmiennej typu **double** i dodasz je do siebie, otrzymana odpowiedź powinna bez problemu zmieścić się w podanej dokładności.)

Ocenianie

Jest 12 podzadań.

Podzadania 1, 2, 3, 4, 5, 6 są warte odpowiednio 4, 8, 12, 12, 12 i 32 punktów. Wszystkie mają $p = 1$, a maksymalne wartości n wynoszą w nich odpowiednio 7, 15, 50, 100, 500 i 2000.

Dla każdego $i \leq 6$, podzadanie $6 + i$ ma $p = 2$, takie samo maksymalne n jak w podzadaniu i, a warte jest $1/4$ punktów podzadania i.

(Łącznie zatem można otrzymać 80 punktów za podzadania z $p = 1$ i 20 punktów za podzadania z $p = 2$.)

Przykłady

Dla danych wejściowych:

```
1
2
4
0 0
1 0
1 1
0 1
5
0 0
8 0
8 10
4 20
0 10
```

poprawnym wynikiem jest:

```
3.000000000000
36.770329614269
```

Dla danych wejściowych:

```
2
2
4
0 0
1 0
1 1
0 1
5
0 0
8 0
8 10
4 20
0 10
```

poprawnym wynikiem jest:

```
3 2
4 1
2 1
4 3
5 3
3 2
2 1
```

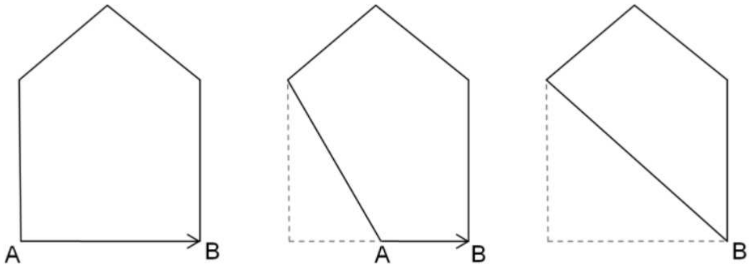
Úloha: POL
Redukcia mnohouholníku

slovak

ONTAK 2022, deň 2. Pamäťový limit: 512 MB. *30.06.2022*

Uvažujme konvexný mnohouholník s n vrcholmi (a teda n stranami). Vykonáte postupnosť $n-1$ operácií, ktorými tento mnohouholník zredukujete na jeden bod.

Operácia pozostáva z výberu dvoch susedných vrcholov A a B, a presunutia A na B. Na konci operácie budú tieto dva vrcholy zlúčené do jedného, čím vznikne mnohouholník s o jedna menším počtom vrcholov. Nový vrchol má stále označenie B (označenie A zmizlo).



Cena operácie sa rovná Euklidovskej vzdialenosti medzi A a B.
Dostanete mnohouholník a vašou úlohou bude vyriešiť jeden z dvoch súvisiacich problémov.

1. Nájdite minimálnu celkovú cenu tejto postupnosti operácií.
2. Nájdite jednu postupnosť $n-1$ operácií s najnižšou celkovou cenou.

Vstup

Každý vstup začína riadkom obsahujúcim číslo $p \in \{1, 2\}$: problém, ktorý budete riešiť.

Druhý riadok vstupu obsahuje počet testov t, ktoré budú nasledovať. (V každom z nich riešite problém p.)

Každý test začína riadkom obsahujúcim číslo n. Nasledujúcich n riadkov obsahuje súradnice x_i, y_i jedného z vrcholov mnohouholníku.

Všetky súradnice sú celé čísla. Všetky vrcholy sú navzájom rôzne. Vrcholy sú uvedené v protismere hodinových ručičiek.

Polygón je zaručene konvexný, ale nie nevyhnutne striktne konvexný – t.j. môže mať viaceré vrcholy ležiace na jednej priamke.

Pre každý vstup platí $t \leq 5$. Pre každý test v každom vstupe platí $n \leq 2000$ a pre koordináty platí $|x_i|, |y_i| \leq 10^6$.

Výstup

Pre každý test vypíšte jeho riešenie.

Ak riešite problém $p = 1$, riešením je jeden riadok s jedným reálnym číslom: minimálne celkové náklady. (Odporúčame vypísať aspoň desať desatinných miest.)

Ak riešite úlohu $p = 2$, riešením je postupnosť $n-1$ riadkov, každý v tvare A_j B_j: pre každé presunutie jeden riadok obsahujúci vrchol, ktorý sa bude presúvať a jeho suseda, na ktorého sa presúva.

Aby sa predišlo chybám pri zaokrúhľovaní, v každom teste bude vaša odpoveď považovaná za správnu, ak sa minimálne celkové náklady vo vašom riešení líšia od referenčného riešenia o najviac 10^{-4}. (Ak vypočítate všetky vzdialenosti s použitím doubles a jednoducho ich sčítate, tak konečná hodnota, ktorú získate, by mala byť vždy bezpečne v tomto rozsahu.)

Hodnotenie

Je 12 sád vstupov:

Sady 1, 2, 3, 4, 5, 6 sú za 4, 8, 12, 12, 12 a 32 bodov. Pre všetky platí $p = 1$ a maximálna hodnota n je 7, 15, 50, 100, 500, a 2000.

Pre každé $i \leq 6$ pre sadu $6 + i$ platí, že $p = 2$, hodnota n je rovnaká ako v sade i, a má hodnotu jednej štvrtiny bodov sady i.

(Celkovo môžte získať 80 bodov za vyriešenie všetkých testov s $p = 1$ a zvyšných 20 za testy s $p = 2$.)

Príklady

Pre vstup:

```
1
2
4
0 0
1 0
1 1
0 1
5
0 0
8 0
8 10
4 20
0 10
```

je správny výsledok:

```
3.000000000000
36.770329614269
```

Pre vstup:

```
2
2
4
0 0
1 0
1 1
0 1
5
0 0
8 0
8 10
4 20
0 10
```

je správny výsledok:

```
3 2
4 1
2 1
4 3
5 3
3 2
2 1
```

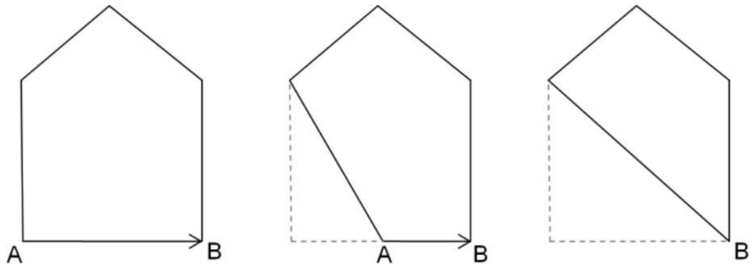
Завдання: POL
Polygon compression

ukrainian

ONTAK 2022, день 2. Обмеження пам'яті: 512 MB. *30.06.2022*

Розглянемо опуклий багатокутник з n вершинами (а отже, n сторонами). Ви збираєтеся виконати послідовність з $n-1$ ходів, які стиснуть багатокутник в одну точку.

Дозволене переміщення складається з вибору двох сусідніх вершин A і B і перетягування A на B. В кінці переміщення дві вершини зливаються в одну, утворюючи багатокутник, у якому на одну вершину менше. Нова вершина все ще має вершину B (вершина A зникла).



Вартість переміщення дорівнює евклідовій відстані між A і B. З огляду на такий багатокутник, вам пропонується розв'язати одну з двох пов'язаних задач:

1. Знайдіть найменшу загальну вартість такої послідовності ходів.
2. Знайдіть одну послідовність з $n-1$ ходів з найменшою загальною вартістю.

Вхідні дані

Кожне введення починається з рядка, що містить число $p \in \{1, 2\}$: задача, яку ви розв'язуєте.

Другий рядок вхідних даних містить t – кількість тестових випадків, які слідують. (У кожному з них ви розв'язуєте задачу p.)

Кожен тестовий приклад починається з рядка, що містить число n. Після цього дано n рядків, кожен з яких містить координати x_i, y_i однієї з вершин багатокутника.

Усі координати цілі числа. Усі вершини різні. Вершини подано в порядку проти годинникової стрілки.

Багатокутник гарантовано є опуклим, але не обов'язково суворо опуклим – тобто допускаються колінеарні вершини.

У кожному тестовому випадку $t \leq 5$, $n \leq 2000$, і всі координати задовольняють $|x_i|, |y_i| \leq 10^6$.

Вихідні дані

Для кожного тесту виведіть своє рішення.

Якщо розв'язуєте задачу $p = 1$, то розв'язком є один рядок з одним дійсним числом: мінімальною загальною вартістю. (Ми рекомендуємо виводити не менше десяти знаків після коми.)

Якщо розв'язуєте задачу $p = 2$, рішення являє собою послідовність з $n-1$ рядків, кожен із яких має вигляд A_j B_j: для кожного переміщення має бути один рядок, що містить вершину, яку потрібно перетягнути, та її сусіда, на яку потрібно перетягнути.

Щоб уникнути помилок округлення, у кожному тестовому випадку ваша відповідь вважатиметься правильною, якщо мінімальна загальна вартість у вашому рішенні відрізняється від еталонного рішення не більше ніж на 10^{-4}. (Якщо ви просто обчислюєте всі відстані як подвійні та додаєте їх разом без будь-яких особливих міркувань, кінцеве значення, яке ви отримаєте, завжди має комфортно знаходитися в цьому діапазоні.)

Оцінювання

Існує дванадцять підзадач.

Підзавдання 1, 2, 3, 4, 5, 6 оцінюються в 4, 8, 12, 12, 12 і 32 бали. Усі вони мають $p = 1$ і максимальне значення n у них становить 7, 15, 50, 100, 500 і 2000.

Для кожного $i \leq 6$ підзадача $6+i$ має $p = 2$, такий самий максимум n, що й підзадача i, і вона коштує чверті балів підзадачі i.

(Усього ви отримуєте 80 балів за рішення всіх тестів з $p = 1$ і решту 20 за тести з $p = 2$.)

Приклади

Розглянемо наступні вхідні дані:

```
1
2
4
0 0
1 0
1 1
0 1
5
0 0
8 0
8 10
4 20
0 10
```

Можливою коректною відповіддю може бути:

```
3.000000000000
36.770329614269
```

Розглянемо наступні вхідні дані:

```
2
2
4
0 0
1 0
1 1
0 1
5
0 0
8 0
8 10
4 20
0 10
```

Можливою коректною відповіддю може бути:

```
3 2
4 1
2 1
4 3
5 3
3 2
2 1
```