Asia-Pacific Informatics Olympiad

APIO 2010
Beijing•China

Saturday, 8 May, 2010

Task Name	commando	patrol	signaling
Time Limit	1 sec	1 sec	2 sec
Memory Limit	64 MB	64 MB	64 MB
Points	100	100	100
Input	`stdin` (keyboard)		
Output	`stdout` (screen)		

Language	*Compiler version*	*Compiler options*
C	`gcc version 4.1.2`	`-m32 -lm`
C++	`g++ version 4.1.2`	`-m32 -lm`
Pascal	`fpc 2.0.4 for i386`	`-Sd -Sh`

Commando

You are the commander of a troop of n soldiers, numbered from 1 to n. For the battle ahead, you plan to divide these n soldiers into several commando units. To promote unity and boost morale, each unit will consist of a contiguous sequence of soldiers of the form $(i, i{+}1, \ldots, i{+}k)$.

Each soldier i has a battle effectiveness rating x_i. Originally, the battle effectiveness x of a commando unit $(i, i{+}1, \ldots, i{+}k)$ was computed by adding up the individual battle effectiveness of the soldiers in the unit. In other words, $x = x_i + x_{i+1} + \cdots + x_{i+k}$.

However, years of glorious victories have led you to conclude that the battle effectiveness of a unit should be adjusted as follows: the adjusted effectiveness x' is computed by using the equation $x' = ax^2 + bx + c$, where a, b, c are known coefficients($a < 0$), x is the original effectiveness of the unit.

Your task as commander is to divide your soldiers into commando units in order to maximize the sum of the adjusted effectiveness of all the units.

For instance, suppose you have 4 soldiers, $x_1 = 2, x_2 = 2, x_3 = 3, x_4 = 4$. Further, let the coefficients for the equation to adjust the battle effectiveness of a unit be $a = -1$, $b = 10$, $c = -20$. In this case, the best solution is to divide the soldiers into three commando units: The first unit contains soldiers 1 and 2, the second unit contains soldier 3, and the third unit contains soldier 4. The battle effectiveness of the three units are 4, 3, 4 respectively, and the adjusted effectiveness are 4, 1, 4 respectively. The total adjusted effectiveness for this grouping is 9 and it can be checked that no better solution is possible.

Input format

The input consists of three lines. The first line contains a positive integer n, the total number of soldiers. The second line contains 3 integers a, b, and c, the coefficients for the equation to adjust the battle effectiveness of a commando unit. The last line contains n integers x_1, x_2, $\ldots$, x_n, separated by spaces, representing the battle effectiveness of soldiers 1, 2, $\ldots$, n, respectively.

Output format

A single line with an integer indicating the maximum adjusted effectiveness achievable.

Sample input

```
4
-1 10 -20
2 2 3 4
```

Sample output

```
9
```

Constraints

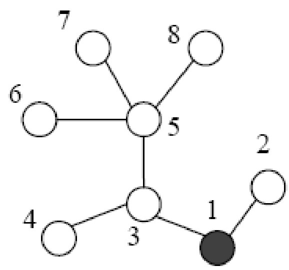
- In 20% of the test cases, $n \leq 1000$;
- In 50% of the test cases, $n \leq 10,000$;
- In 100% of the test cases, $n \leq 1,000,000$, $-5 \leq a \leq -1$, $|b| \leq 10,000,000$, $|c| \leq 10,000,000$ and $1 \leq x_i \leq 100$.

Patrol

In a city, there are N villages numbered 1, 2, ..., N. There are $N-1$ roads connecting them. Each road connects exactly 2 villages, and from any village, one can reach any other village using these roads. The length of each road is 1 unit.

To ensure safety of the people in the city, each day a city police patrol has to travel on every road. The police station is at village 1, so the patrol has to start from village 1 and finally return to village 1 at the end of the day.

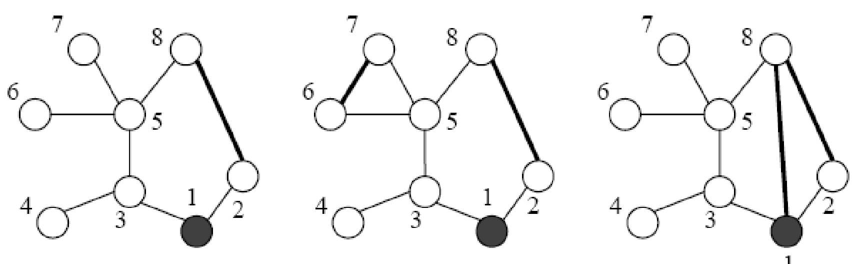
Consider the following example of a city with 8 villages below. Villages are shown as circles and village 1 is shown as a black circle. The roads are the lines connecting these villages. To traverse all roads, the patrol has to travel 14 units each day. Note that the patrol must travel along each road *twice* to complete each day's job.



To reduce the total distance required, the city plans to build K new *shortcuts* between these villages. Each shortcut can connect any two villages. Two shortcuts can end at the same village (see example (c) below). A shortcut can even be a loop; that is, connect a village to itself.

Funding is limited, so K is either 1 or 2. Also, to make sure that the city is not wasting money, it is required that the patrol must travel along each shortcut *exactly once a day*.

Consider the following examples.



(a) (b) (c)

In example (a), one shortcut is built, and the total distance is 11. In example (b), two shortcuts are built, and the total distance the patrol has to travel is 10. In the last example (c), two shortcuts are built, however since there is a requirement on the number of times the patrol can travel on each shortcut exactly once, the total distance now becomes 15.

Write a program that reads the information about the roads between the villages and the number of shortcuts to be built and computes the location for the shortcuts that minimizes the total distance the patrol has to travel each day.

Input format

The first line of input contains two integers N and $K(1 \leq K \leq 2)$. The next $N-1$ lines contain information about the roads. Each of these lines contains two integers A and B, $(1 \leq A, B \leq N)$, which says that there is a road connecting village A and village B.

Output format

Your program should output one line with an integer which is the minimum distance the patrol has to travel after K shortcuts have been built.

Sample input 1

```
8 1
1 2
3 1
3 4
5 3
7 5
8 5
5 6
```

Sample output 1

```
11
```

Sample input 2

```
8 2
1 2
3 1
3 4
5 3
7 5
8 5
5 6
```

Sample output 2

```
10
```

Sample input 3

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

Sample output 3

```
6
```

Constraints

- In 10% of the test cases, $N \leq 1,000$ and $K = 1$.
- In 30% of the test cases, $K = 1$.
- In 80% of the test cases, the maximum number of adjacent villages for each village is at most 25.
- In 90% of the test cases, the maximum number of adjacent villages for each village is at most 150.
- In 100% of the test cases, $3 \leq N \leq 100,000$ and $1 \leq K \leq 2$.

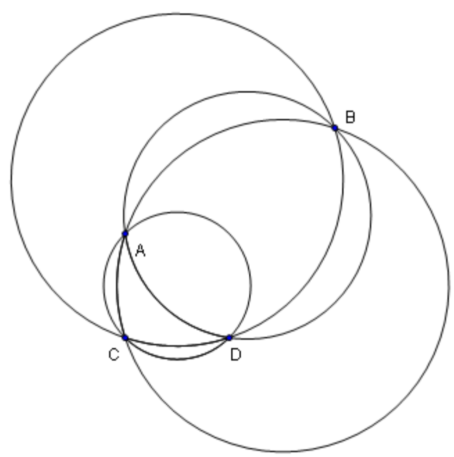
Signaling

A telecom company is developing a GSM network in the city of Beijing. There are n houses in the city that need to be covered by the network. Due to budget constraints, the company can install only a single antenna.

To simplify the placement of this antenna, the location will be determined by picking 3 of the n houses to make a circle and then placing the antenna at the center of this circle. The range of the antenna will be such that all houses that lie within this circle, including those on the boundary of the circle, are covered.

The company plans to pick the 3 houses at random, so they want to compute the average number of houses that will be covered across all possible choices for the location of the antenna.

For example, suppose there are 4 houses, A, B, C and D located as shown in the figure below.



If we choose the circle defined by ABC or BCD, every house is covered. If we choose the circle defined by ACD or ABD, the fourth house is not within the circle covered by the antenna. Therefore, the average number of houses covered is $\frac{1}{4}(4 + 4 + 3 + 3) = 3.5$.

Your task is to compute the average number of houses covered by the signal, given the locations of the houses. The positions of the houses are given in terms of a 2-dimensional coordinate system in which all houses have integer coordinates. You are guaranteed that no three houses lie on the same line and no four of them lie on the same circle.

Input format

The first line of input contains a single positive integer n, the total number of the houses. This is followed by n lines, describing the locations of the houses. For $i \in \{1, 2, \ldots, n\}$, the coordinates of house i are given by a pair of integers x_i and y_i on line $i+1$ of the input, separated by spaces.

Output format

The output should contain a single real number, the average number of houses that are be covered by the signal. The absolute error of the result should be less than or equal to 0.01.

Sample input

```
4
0 2
4 4
0 0
2 0
```

Sample output

```
3.500
```

Sample Explanation

$3.5, 3.50, 3.500, \cdots$ are all considered correct outputs, moreover, 3.51, 3.49, 3.499999, $\cdots$ are also acceptable.

Constraints

- In 100% of the test cases, for $i \in \{1, 2, \ldots, n\}$, the coordinates (x_i, y_i) of house i are both integers such that $-1000000 \le x_i, y_i \le 1000000$. No three houses are located on a line and no four houses are located on a circle.
- In 40% of the test cases, $n \le 100$.
- In 70% of the test cases, $n \le 500$.
- In 100% of the test cases, $3 \le n \le 1,500$.