



2010 年 5 月 8 日 (星期六)

	任務		
	COMMANDO	PATROL	SIGNALING
時限	1 秒	1 秒	2 秒
記憶體上限	64 MB	64 MB	64 MB
分數	100	100	100
輸入	準標輸入 (鍵盤輸入)		
輸出	準標輸出 (螢幕輸出)		

語言	編譯器版本	編譯器參數
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

你是一位部隊司令。你的部隊中共有  $n$  位士兵，他們分別編號從 1 到  $n$ 。面對未來的戰鬥，你打算將這  $n$  個士兵分割成為數個突擊隊單元。為了促進團結和鼓舞士氣，每個單元將由編號相連的士兵，如  $(i, i+1, \dots, i+k)$ ，所組成。

每個士兵  $i$  都有一個作戰效能值  $x_i$ 。本來，一支由士兵  $(i, i+1, \dots, i+k)$  所組成的突擊隊單元的戰鬥效能總值  $x$  是將突擊隊中每個士兵的作戰效能值加起來所得的總和。換句話來說為  $x = x_i + x_{i+1} + \dots + x_{i+k}$ 。

然而，多年來的光輝勝利使你得出這樣的結論，就是一個單元的作戰效能總值應作如下調整：調整後的效能總值  $x'$  是通過以下方式計算出來  $x' = ax^2 + bx + c$ ，其中  $a, b, c$  是已知的係數 ( $a < 0$ )， $x$  為原單元的效能總值。

作為指揮官，你的任務是將部隊的士兵劃分成突擊隊單元，以求得到調整後的單元效能總值的最大可能總和。

舉例說，假設你有 4 名士兵， $x_1 = 2$ ， $x_2 = 2$ ， $x_3 = 3$ ， $x_4 = 4$ 。此外，設用以調整一個單元的作戰效能總值的方程的係數為  $a = -1$ ， $b = 10$ ， $c = -20$ 。在這種情況下，最好的辦法是將部隊劃分為三個突擊隊單元：第一單元包含士兵 1 和 2，第二個單元中有士兵 3，第三個單元中有士兵 4。這三個單元的原本作戰效能總值分別為 4，3，4。而調整後的作戰效能總值則為 4，1，4。而調整後的作戰效能總值的總和為 9，除這個方案外就沒有其他更好的解決方案了。

### 輸入格式

輸入包含三行。第一行包含一個正整數  $n$ ，它代表士兵的總人數。第二行包含 3 個整數  $a$ ， $b$  和  $c$ ，它們是一個突擊隊單元的作戰效能總值調整方式的係數。最後一行有  $n$  個整數，它們分別順序為  $x_1$ ， $x_2$ ， $\dots$ ， $x_n$ 。每個整數之間以一個空格分開。

### 輸出格式

輸出應只有一行，其上只有一個整數，它代表可以求得的調整後的作戰效能總值的最大總和。

輸入樣例	輸出樣例
4	9
-1 10 -20	
2 2 3 4	

## 限制條件

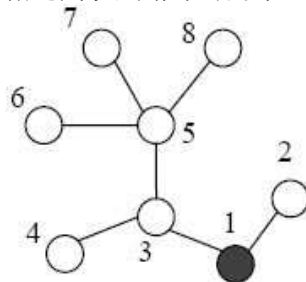
- 在 20% 的測試數據中， $n \leq 1,000$ ;
- 在 50% 的測試數據中， $n \leq 10,000$ ;
- 在 100% 的測試數據中， $n \leq 1,00,0000$ ， $-5 \leq a \leq -1$ ， $|b| \leq 10,000,000$ ， $|c| \leq 10,000,000$  且  $1 \leq x_i \leq 100$ 。

## Patrol

在一個城市裡有  $N$  個村莊分別編號為  $1, 2, \dots, N$ 。在那兒有  $N - 1$  條道路把這些村莊連接起來。每條道路連接正好兩個村莊，而從任何一個村莊，我們均可以通過這些道路到達任何另外一個村莊。每條道路的長度均為 1 個單位。

為確保城市人民的安全，每天市警察均要巡邏每一條道路。而警局就在 1 號村莊內，所以巡邏的路線必須從 1 號村莊開始，而且最後返回 1 號村莊作為一天的結束。

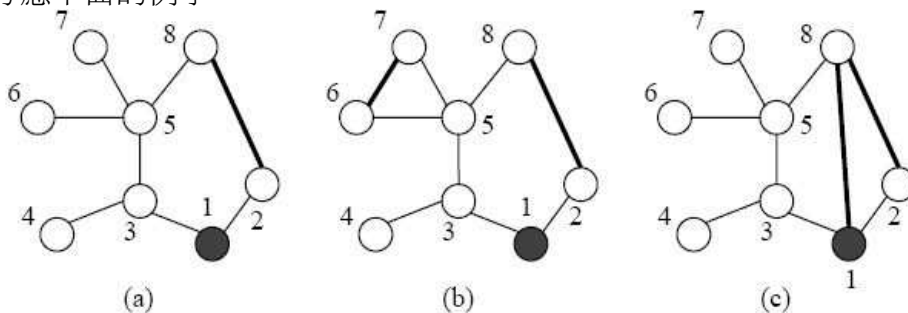
考慮下面的例子，一個城市中有 8 個村莊。每個村莊在下圖以一圓圈表示，而 1 號村莊則以一個黑色圓圈表示。而道路則是連接這些村莊的線。為了遍歷所有道路，每天巡邏時都要走過 14 個單位的路程。請注意，在這裡巡邏時必須沿每條道路走兩次才能完成每一天的工作。



為減少巡邏總路程要求，城市規劃打算在村莊之間建設  $K$  條新捷徑。每條捷徑可以連接任何兩個村莊。兩條捷徑可以結束在同一條村莊（見下例 (c)）。捷徑甚至可以是一個環，即其兩端連接到同一個村莊。

但資金是有限的，所以  $K$  的值只可以是 1 或 2。此外，為確保城市並不浪費錢，每日巡邏時都要求必須走過每條捷徑正好一次。

考慮下面的例子。



在例 (a) 中，建立了一條捷徑，總巡邏路程為 11 個單位。在例 (b) 中，建立了兩條捷徑，總巡邏路程為 10 個單位。而最後一個例子 (c)，同樣是建立了兩條捷徑，但由於規定在巡邏時必須經過每條捷徑正好一次，所以總巡邏路程為變成 15 個單位。

請編寫一個程序，以讀取村莊間的的道路資料以及所要建的捷徑數目，然後計算出能使每天總巡邏路程變為最少的捷徑建設方案。

## 輸入格式

第一行輸入包含兩個整數  $N$  和  $K$  ( $1 \leq K \leq 2$ )。其後的  $N - 1$  行每行包含關的道路的資料。這部份中，每行包含兩個整數  $A$  和  $B$ ，( $1 \leq A, B \leq N$ )，它表示有一條將村莊  $A$  和  $B$  連接的道路存在。

## 輸出格式

你的程序應輸出一行資料，其上只有一個整數，它代你所找到的利用  $K$  條捷徑而得的最短巡邏的旅程。

輸入樣例 1	輸出樣例 1
8 1 1 2 3 1 3 4 5 3 7 5 8 5 5 6	11

輸入樣例 2	輸出樣例 2
8 2 1 2 3 1 3 4 5 3 7 5 8 5 5 6	10

輸入樣例 3	輸出樣例 3
5 2 1 2 2 3 3 4 4 5	6

## 限制條件

- 在 10% 的測試數據中， $N \leq 1,000$  且  $K = 1$ 。
- 在 30% 的測試數據中， $K = 1$ 。
- 在 80% 的測試數據中，對於每個村莊而言，與其相連的村莊數目最大為 25。
- 在 90% 的測試數據中，對於每個村莊而言，與其相連的村莊數目最大為 150。
- 在 100% 的測試數據中， $3 \leq N \leq 100,000$  且  $1 \leq K \leq 2$ 。

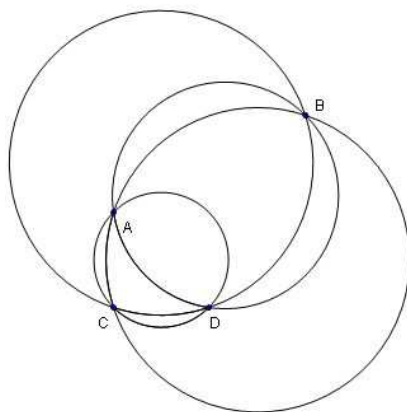
## Signaling

一家電信公司正在北京市開發一個 GSM 網絡。在內共有  $n$  家用戶需要網絡覆蓋。由於預算的限制，該公司只能在內安裝一個天線。

為了簡化這個天線放置的位置問題，天線放置的方法是在內的  $n$  家用戶中選 3 家，以這 3 家的位置作一圓，而天線就會放在這個圓的中心。這樣，天線的網絡覆蓋範圍就會包括位於圓上及圓內的所有用戶。

該公司計劃選擇以隨機的方式選出 3 家用戶，所以他們想計算對於所有不同可能的天線選擇位置，其用戶覆蓋的平均值。

例如，假設有目前有 4 家用戶， $A$ ， $B$ ， $C$  和  $D$ ，其所在位置如下圖所示。



如果我們選以  $ABC$  或  $BCD$  來畫出的圓，它們都能覆蓋內所有的用戶。如果我們選以  $ACD$  或  $ABD$  來畫出的圓，則內第四家用戶未能被覆蓋到。因此，覆蓋能力的平均值為  $\frac{1}{4}(4 + 4 + 3 + 3) = 3.5$ 。

你的任務是在給定每家用戶的位置情況下，計算天線覆蓋能力的平均值。用戶的位置是以二維座標系統來給定的，而所有座標值均為整數。可以保證的是沒有任何三家用戶的位置是在一條直線上，而且沒有任何四家用戶位置會在用一個圓上。

## 輸入格式

第一行輸入包含一個正整數  $n$ ，它代表用戶的總數。隨後的  $n$  行，每行描寫一個用戶的位置。即對於  $i \in \{1, 2, \dots, n\}$ ，在第  $i + 1$  行的輸入上有兩個整數，它們以一個空格分開，它們代表著第  $i$  個用戶所在的  $x_i$  及  $y_i$  座標。

## 輸出格式

輸出應包含單一個實數，它代表天線信號覆蓋能力的平均值。這個數字的絕對誤差應少於或等於 0.01。

輸入樣例	輸出樣例
4	3.500
0 2	
4 4	
0 0	
2 0	

## 例子說明

上例中，3.5，3.50，3.500，... 等均為正確答案，同時答案如 3.51，3.49，3.499999 亦均可以接受為正確答案。

## 限制條件

- 在 100% 的測試數據中，對於每個用戶  $i$  的座標  $(x_i, y_i)$  ( $i \in \{1, 2, \dots, n\}$ )，它們均為整數且  $-1000000 \leq x_i, y_i \leq 1000000$ 。沒有任何三家用戶的位置是在一條直線上，而且沒有任何四家用戶位置會在用一個圓週上。
- 在 40% 的測試數據中， $n \leq 100$ 。
- 在 70% 的測試數據中， $n \leq 500$ 。
- 在 100% 的測試數據中， $3 \leq n \leq 1,500$ 。