二階

時間限制: 1 秒

記憶體限制: 256 MiB

Brianlee 是個學霸，不僅每天認真讀書，也會把握時間精進程式設計能力。最終他不僅進入了奧林匹亞選訓營一階，更進入了台大資工二階！

台大資工二階的算分方式如下：

學測、英聽篩選方式			甄選總成績採計方式及佔總成績比例				
第一階段			第二階段				
科目	檢定	篩選倍率	學測成績採計方式	佔甄選總成績比例	指定項目	檢定	佔甄選總成績比例
國文	前標	--	--	20%	審查資料	70分	40%
英文	頂標	10	*1.25		數學筆試或程式設計	--	40%
數學	頂標	3	*2.00				
社會	--	--	--				
自然	頂標	10	*1.00				
--	--	--	--				
英聽	--	--	--				

學測成績計算方式為： $20 \times \dfrac{\text{英文級分} \times 1.25 + \text{數學級分} \times 2 + \text{自然級分} \times 1}{15 \times 4.25}$

審查資料、程式設計部分為： $\text{原始分數} \times \dfrac{40}{100}$。

一個考生的總分是每項分數的總和。

你已經先知道了台大資工二階的錄取門檻，BrianLee 會告訴你他各科的成績，請幫助 BrianLee 判斷他是否會錄取台大資工。

不須考慮 BrianLee 未通過一階的情形，因為他是學霸。

Input

第一行是三個小於等於 15 的非負整數，分別代表 BrianLee 的英文、數學、自然學測級分。

第二行是兩個小於等於 100 的浮點數，分別代表 BrianLee 的審查資料、程式設計成績。

第三行是一個小於等於 100 的浮點數，代表台大資工二階的錄取門檻。

所有浮點數小數以下位數不超過 6 位。

Output

如果BrianLee可以錄取台大，請輸出"YA"(不含引號)。

如果BrianLee無法錄取台大，請輸出"QQ"(不含引號)。

Scoring

Subtask	Score	Constraints
1	40	*BrianLee* 的學測滿級分,且各部分成績以及錄取門檻都是整數。
2	35	*BrianLee* 的學測滿級分。
3	25	無額外限制。

Examples

input
15 15 15 85 100 100
output
QQ

input
15 15 15 5.73323 80.4892 65.4782
output
QQ

input
15 8 5 75.9602 43.6891 49.0542
output
YA

Note

範例測資三解釋：

第三筆範例測資中，BrianLee 的原始分數為 $12.4706... + 30.3841... + 17.4756...$ ， 高於門檻分數 49.0542 。

國手

時間限制: 1 秒
記憶體限制: 256 MiB

8e7 當上國手了！他憑著過人的實力和天才般的考試技巧達到目標。

但是許多人不知道的是，8e7 有個神秘的絕招！為了不讓其他人發現他的真實力，8e7 在每場模考之前，可以選擇要裝弱或是正常發揮，而且他知道裝弱和正常發揮分別可以獲得幾分。

由於 8e7 也是個先知，他從第一場模考就知道最後國手線的分數了（至少要幾分才能是國手）。因此這時 8e7 想要知道，他最多可以在幾場模考裝弱，到最後仍在國手線上？在國手線上的定義為，N 次模考的分數和 $\geq$ 國手線分數。

對 8e7 來說，這題實在太簡單了，所以他把這題委託給您了，你能寫一個程式計算出答案嗎？

Input

第一行有兩個數字 N, K ，分別代表有幾場模考和國手線的分數。

第二行有 N 個數字，第 i 個數字為 8e7 在第 i 次模考裝弱時可拿的分數 a_i。

第三行有 N 個數字，第 i 個數字為 8e7 在第 i 次模考正常發揮時可拿的分數 b_i。

Output

輸出一個整數，代表 8e7 最多可以裝弱的天數。

由於 8e7 太強了，因此他在每場都正常發揮下一定能當國手。

Scoring

- $1 \leq N \leq 2 \times 10^5$
- $1 \leq K \leq 10^{12}$
- $0 \leq a_i \leq b_i \leq 10^6$

Subtask	Score	Constraints
1	12	$N \leq 100$
2	26	$N \leq 5000$,且 $8e7$ 裝弱時必定拿0分。
3	13	$N \leq 5000$
4	27	$8e7$ 裝弱時必定拿0分
5	22	無額外限制

Examples

input

```
2 200
100 150
101 160
```

output

```
2
```

input

```
4 495
135 110 40 100
```

```
140 130 65 186
```

output

```
2
```

input

```
1 100
99
123
```

output

```
0
```

Note

範例測資二解釋：

8e7:可以在第1、2天裝弱，總分為 $135 + 110 + 65 + 186 = 496$。

由於本題的輸入量十分龐大，如使用C++的cin讀入資料，請務必於main函式開頭加上
ios::sync_with_stdio(0),cin.tie(0); 以避免因輸入過慢而TLE。

鐵人三項

時間限制: 1 秒

記憶體限制: 256 MiB

熱愛運動的 ub33 最喜歡鐵人三項了，在選擇選修課時毫不猶豫地選擇了它！今天，ub33 打算參加梯歐哀盃鐵人三項大賽，這項比賽的規則如下：

選手必須按照游泳、腳踏車、跑步的順序完成比賽，且每一種運動方式都需至少在一路段上使用，不可任意改變順序，也不可穿插 (一旦從游泳轉換為腳踏車，便不可在接下來的路段游泳)。但不限定各路段使用的方式，也就是說，在水上跑步，柏油路上游泳，都是符合比賽規則的！

ub33 事前探勘過場地，知道每段路以三種運動方式所需要的時間，你能幫 ub33 算出，他最少需要多少時間才能完賽嗎?

Input

每筆測資的第一行是一個正整數 N ，代表梯歐哀盃鐵人三項大賽有 N 個路段。第 2 到 $N+1$ 行各有 3 個正整數 a_i, b_i, c_i ，依序為以游泳、腳踏車、跑步方式通過該路段所需時間。

Output

輸出一整數，代表 ub33 最小完賽時間。

Scoring

- $3 \leq N \leq 2 \times 10^5$
- $1 \leq a_i, b_i, c_i \leq 10^9$

Subtask	Score	Constraints
1	23	$N \leq 300$, $1 \leq a_i, b_i, c_i \leq 10^5$
2	24	$N \leq 5000$
3	26	每個路段中,游泳所需時間皆為 10^9
4	27	無額外限制

Examples

input

```
3
2 3 4
5 10 3
6 8 9
```

output

```
21
```

input

```
7
2 3 4
5 10 3
6 8 9
5 9 1
4 2 3
4 6 2
7 5 3
```

output
24

Note

範例測資二解釋：

於1-2路段游泳，第3路段騎腳踏車，第4-7段跑步，總時間花費： $2+5+8+1+3+2+3=24$ 。

由於本題的輸入量十分龐大，如使用C++的cin讀入資料，請務必於main函式開頭加上
ios::sync_with_stdio(0),cin.tie(0); 以避免因輸入過慢而TLE。

舒服

時間限制: 1 秒

記憶體限制: 256 MiB

joylintp 是舒服大師！只要 joylintp 一發文預測，全世界都為之膽戰心驚。小至全國賽、北市賽，大至 ICPC、IOI，無人能逃出 joylintp 魔掌。

繼承了 joylintp 力量的你，偶然在房間不起眼的角落，翻出了一本黑色精裝的厚重書本，封面畫著排滿未知名字和幾何圖形的魔法陣，上面以暗紅色和血書般的字體寫著《舒服魔導書》。好奇翻開一看，每頁密密麻麻寫著一堆數字、字母和記號，但都不超出 ASCII 32 至 126 的範圍。

這如同加密過的內容引起了你極大的興趣，裡面肯定是些 joylintp 的舒服名單。你決定解出它的內文，以在下次 joylintp 發文前提醒舒服名單上的人封鎖 joylintp。

經過八八六十四天不眠不休的研究，你發現每個段落都是由一串文字 S 作為開頭，接下來是一長串十進位數字 A 。

你發現，A 的位數必定是 10 的倍數，將 A 每 10 位數字切成一段，會得到一序列 B。將 B 中的每個數字分別轉換為長度 32 的二進位整數後按順序全部串接在一起，得到二進位整數 C。就可以開始將 C 轉換為原始文章了！

S 的長度 k 代表 C 必須被轉換成 $k+1$ 進位，而 S 中的第 i 個字 S_i 代表著 C 在轉換為 $k+1$ 進位表達後，數字 i 必須被替換為 S_i 方可得出原文。保證 $k+1$ 必定為 2 的正整數次方。

附帶一提， $k+1$ 進位表達中，每個位數的 0 不論在哪一對 (S, A) 中，都應被轉換為換行記號 $\mathrm{\backslash n}$ ，而換行不會出現在 S 中。

不過你覺得人工翻譯太累了，萬一一個算錯，到時害 joylintp 被一堆人封鎖豈不尷尬。你決定以機關術創造一隻使魔來解密 joylintp 的《舒服魔導書》。

Input

輸入共有兩行，第一行包含一個可能含空白的字串 S，第二行包含一個只含有 0-9 的字串 A。

Output

輸出解密後的原文，可能包含多行。如開頭有多餘換行，請去除。不論原文結尾是否是換行，都應該在原文之後額外輸出一個換行。

Scoring

- 保證若正確將 A 轉換為 B,B 中的每個數字皆會小於 2^{32}
- $32 \leq S_i < 127$
- $1 \leq k < 128$
- $0 < len(A) \leq 10^5$
- 對任意 k 必存在一正整數 x 滿足 $k = 2^x - 1$

Subtask	Score	Constraints
1	1	範測
2	9	A 的長度 ≤ 10、S 不包含空白、原文不包含換行
3	10	A 的長度 ≤ 10
4	15	$k = 1$, A 的長度 ≤ 1000
5	25	k 只會是 $1, 3, 7$, A 的長度 ≤ 1000
6	30	A 的長度 ≤ 1000
7	10	無特別限制

Examples

input
dmohT2y_Wxi1M0ulvqnDZ-sg7eK9Yab 0110614332070455091539205807261483610117045747252620137546833956475637036576183128104512
output
2qbingxuan daniel071292 ToMmyDong Yensean Kevin_Zhang_TW

input
kaobi;E6?d/lr-CsHxP0t2NhecRqOZ2 000106806517789664540080134452296105017700316791183558873449196519740
output
balbit HNO2 PolarisChiba ColdEr66 ericxiao

input
4 0000000682
output
4 4 4 4 4

Note

範例測資三解釋：

將 0000000682 每 10 個位數轉換成一整數後，將其組成一序列，得到 682 ，將序列中每個整數分別轉換為長度為 32 的二進位整數，得到 00000000000000000000001010101010。

依照 S 轉換並去除前導 0，得到 $4\backslash n4\backslash n4\backslash n4\backslash n4\backslash n$，最後多加一個換行。

火力全開

時間限制: 1 秒

記憶體限制: 256 MiB

Judge 已經火力全開了!

身為一名法官，住在小鎮門牌號碼 1 號的 WiwiHo 每天都火力全開的處理各式案件，小鎮裡的人們也都喜歡找 WiwiHo 解決紛爭。然而，由於做出一次公正的裁決需要約 30 秒的時間，等待的隊伍總是大排長龍。

作為鎮長的你，決定重新安排小鎮路上的道路。然而，由於經費有限，你只打算以最少的道路，使每戶間都能直接或間接的相鄰。更準確來說，在 N 位居民的小鎮，你會興建 $N-1$ 條路，使整個小鎮的連接關係形成一樹狀結構。

當一個人千里迢迢來到 WiwiHo 家時，會累積相當於來到 WiwiHo 家距離的不滿值，若 judge 結果不合預期，這股不滿將會全部爆發。由於無法提前知道 judge 結果，你必須盡力使可能出現的最大不滿值最小。也就是說，必須使離 WiwiHo 最遠的居民的住家，離 WiwiHo 盡可能的近。

Input

每筆測資的第一行是兩個正整數 N, M，代表小鎮有 N 位居民及 M 個可能的道路興建方案。

接下來的 M 行有三個正整數 a_i、b_i、c_i ，代表居民 a_i 和 b_i 的家之間可以興建一條長度為 c_i 的路。

Output

輸出一個整數，代表在所有可能的興建方案中，最小的最大不滿值。

Scoring

- $2 \le N \le 10^5$
- $N-1 \le M \le \min(\ 2 \times 10^5,\ \frac{N \times (N-1)}{2}\ )$
- $1 \le a_i, b_i \le N$
- $a_i \neq b_i$
- $0 \le c_i \le 10^9$
- 保證至少有一種興建方法,使得所有人都能**夠**直接或間接的抵達WiwiHo家。

Subtask	Score	Constraints
1	33	$M = N - 1$
2	13	$N \le 3000, M \le 6000$
3	31	$c_i = 1$
4	23	無額外限制

Examples

input
3 2 1 2 5 1 3 7

output
7

input
5 7

```
4 5 11
2 4 12
3 4 7
1 3 4
1 2 5
3 5 8
2 3 10
```

output

```
12
```

Note

範例測資二解釋：

興建第 3、4、5、6 條路，離 WiwiHo 家最遠的 5 號居民家距離為 12。

由於本題的輸入量十分龐大，如使用C++的cin讀入資料，請務必於main函式開頭加上
ios::sync_with_stdio(0),cin.tie(0); 以避免因輸入過慢而TLE。

氣球

時間限制: 1 秒

記憶體限制: 256 MiB

2020 年的 TOI 二階線上選訓營，一款名為 BTD Battles 的遊戲征服了建中。沉默的線上課程中，一顆顆氣球正迅速的朝著對手衝去。

BTD Battles 的規則相當簡單，玩家的目標就是以防禦建築擊破不斷湧來的氣球，並向對手發送氣球使其防禦不住以贏得遊戲。

對於每個種類的氣球，有著他的價格 c_i 以及對防禦建築的壓力指數 p_i 。你可以花費 c_i 元送出該種氣球，並向對手的防禦建築施加 p_i 的壓力。每種氣球在每一回合**只能被發送一次**，防禦設施受到的壓力為當前回合送出的每顆氣球之壓力指數和。

奇妙的是，氣球有著特殊的壓縮機制，你可以花 K 元壓縮一種氣球，使得其價格與壓力指數以任意等比例縮小。舉例來說，若一顆氣球原價為 10，壓力指數為 50，則將其壓縮為價格 5 的氣球後，其壓力指數為 25；將其壓縮為價格 8 的氣球後，其壓力指數為 40。為方便計算，保證壓力指數一定是價格的倍數。

casperwang 現在有一些錢，他想要在這一回合對對手的防禦建築施加盡量大的壓力，你能幫他嗎？

Input

每筆測資的第一行是三個正整數 N, M, K。代表有 N 種氣球，且 casperwang 現在有 M 元，壓縮一顆氣球的成本是 K 元。

接下來的 N 行，每行有兩個數字 c_i, p_i ，代表第 i 種氣球的價格及壓力指數。

Output

輸出一整數，代表這一回合 casperwang 最高可以對敵方防禦建築施加多少壓力。

Scoring

- $1 \leq N, M \leq 3000$
- $0 \leq K \leq 10^9$
- $1 \leq c_i \leq 6000$
- $0 \leq p_i \leq 10^9$
- 保證 p_i 能被 c_i 整除

Subtask	Score	Constraints
1	12	$N \leq 20$
2	10	$N \leq 100$
3	24	$K = 0$
4	26	$K = 10^9$
5	28	無額外限制

Examples

input
5 10 1 2 50 3 45 7 70 1 23 2 44

output

```
172
```

input

```
4 10 7
2 12
7 35
5 5
6 12
```

output

```
47
```

input

```
1 5 500
10 100
```

output

```
0
```

Note

範例測資一解釋:

花費 1 元，將第三種氣球壓縮為價格 1，壓力指數 10 的氣球後，選擇所有氣球。總壓力指數為 $50 + 45 + 10 + 23 + 44 = 172$。

AIQQQ賺多少

時間限制: 1 秒

記憶體限制: 256 MiB

Lipro 在選訓營學習 DP(Double Play) 優化時，發現了 TIOJ(The Institution of Jizz) 上的一題: AI666 賺多少。此題是 106 年高級中學資訊學科能力競賽決賽的一道題目，題目敘述如下。

商品價格經常是起起伏伏，例如石油的價格幾乎時時都有變動，有時上漲，有時下跌。商品交易商在低價時買進高價賣出就可以利用其中的價差獲取利益。

2066 年 6 月 6 日 Automatic Investment 公司以複雜的人工智慧技術開發一套商品價格的預測系統，此系統命名為 $AI-666$，但發展了這麼多複雜的運算技術後，他們現在剩下一個小問題：假設 $AI-666$ 的價格預測是準確的，那麼最多可以在這一段期間賺到多少錢。公司的研發經理希望以這個問題來考驗你，看看你是否有資格加入該公司的研發團隊。

商品交易的規則是這樣的：

1. 只能先買後賣，不可以先賣後買。

2. 每次買與賣都限定是一個單位的商品。同時，在買入之後，賣出之前，不可以再買入。

3. 由於法令的規定，在此期間內最多只能進行 K 次的交易(一次交易包含買賣各一次)。

輸入的資料是 $AI-666$ 系統所預測 N 個時間點的商品價格以及一個正整數 K，請計算不超過 K 次交易的條件下最大可能獲得的利益。舉例來說，以下資料是 11 個時間點的價格：

時間點	0	1	2	3	4	5	6	7	8	9	10
價格	100	90	185	120	80	150	140	180	110	150	50

如果 $K=1$，最大的獲利方式是於第四天以 80 元買進，第七天以 180 元賣出，可以獲利 100 。

如果 $K=2$，最大獲利方式是 90 買進， 185 賣出，然後 80 再買進，180 賣出，總共可以獲利
$(185-90)+(180-80)=195$。

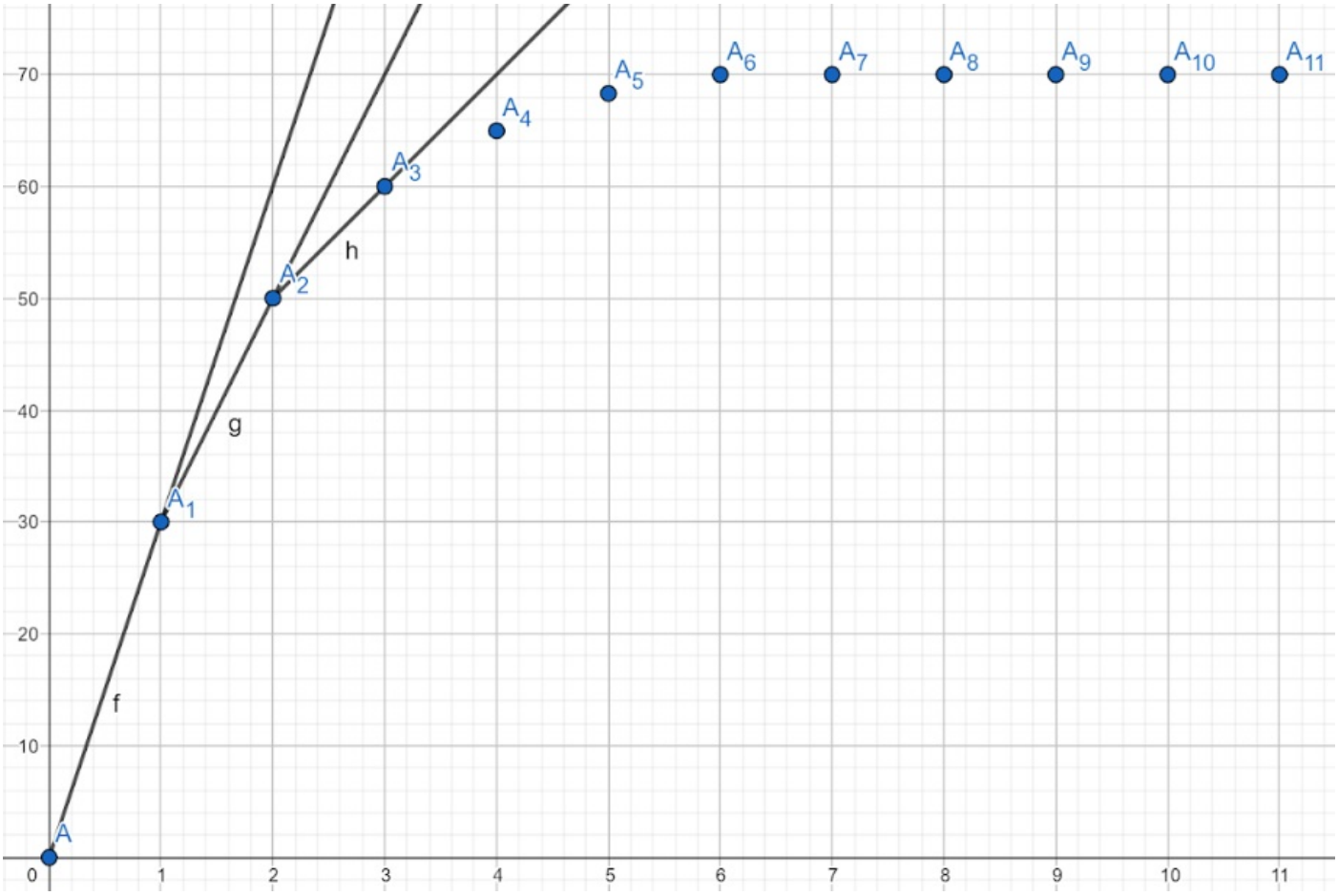
如果 $K=5$，雖然可以交易五次，但交易四次就可以達到最大獲利
$(185-90)+(150-80)+(180-140)+(150-110)=245$。

Lipro 知道他可以用 N,K 進行 DP，以 $dp[i][j]$ 表示在前 i 個時間點中，交易 j 次所能得到的最大價值，枚舉上一次買入的時間點 k，令 $p[i]$ 代表第 i 天的價格，以

$$dp[i][j]=\max(\max_{k<x}(\,p[i]-p[k+1]+dp[k][j-1]\,),\,dp[i-1][j]\,)$$

的方式進行轉移，其中透過一些方法，能夠讓轉移的複雜度降到 $O(1)$ ，通過 $N=50000, K=100$ 的 37 分Subtask。然而，本題想要得到滿分， N,K 都高達 2×10^6 。顯然一般的 DP 不可能解決這個問題。

為了 AC 這題，Lipro 想出了一個神奇的解法 : Lipro 二分搜！ 以 K 為 X 軸，答案為 Y 軸作圖，會得如下圖般的圖形。



透過觀察可以知道，每次增加交易次數時，得到的利潤必定不會遞增。

也就是說，若令 $f(x)$ 為 $K = x$ 時的最高利潤，對於所有 $N \geq 1$，$f(N) - f(N-1) \geq f(N+1) - f(N)$。

從圖形的角度來看，令 $m(x)$ 為線段 x 的斜率，即代表著 $m(f) \geq m(g) \geq m(h)$。

由這個性質，Lipro 推導出了以下結論: 若是交易需要手續費，則隨著手續費的提高，最佳交易次數一定會變少，當多進行一次交易能賺到的錢，還比需要的手續費少，聰明的商人當然不可能進行該次交易！

此外，Lipro 還有著強大的 DP 能力，他發現，當不限制交易次數時，他就能夠在 $O(N)$ 的時間內，知道在 C 元的手續費下，該進行幾次交易是最好的。作法跟上述相近，但這次，他並不需要代表現在交換次數的維度了。只需以 $dp[i]$ 表示在前 i 個時間點中，不限交易次數所能得到的最大價值，枚舉上一次買入的時間點 k，以

$$dp[i] = \max(\max_{0 \leq k < i-1} (p[i] - p[k+1] + dp[k] - C) , dp[i-1])$$

進行轉移，並同時紀錄交易次數，就可以得到答案！

同樣的，透過一些技巧，複雜度就可以降到 $O(N)$！ 結合以上兩點，Lipro 就可以解決這個限制交易 K 次的問題了。 方法如下：

二分搜手續費 C ，計算以 C 為手續費時的最大利潤 p 以及達到這個利潤所需的交易次數 q 。假設以當前 C 進行 DP 的最佳交易次數 $q > k$，則調高手續費，否則調降手續費。最終，當最佳交易次數為 k 時，將利潤 p 加上手續費 $C \times k$ ，就可以得到在限制 k 次交易下，最大的利潤了！

Lipro 已經寫好了二分搜的過程，現在他想請你幫他算出，在一次交易(買+賣合起來算一次)手續費為 C 的情況下，他最多可以賺到多少利潤，以及得到此利潤所需進行的最小交易次數，以讓他可以順利解決整個問題！

Input

每筆測資共有二行。第一行為兩個正整數 N 與 C。分別代表時間點數與交易一次(買+賣)所需手續費。

第二行有 N 個以空白間隔的正整數，第 i 項 p_i 代表在第 i 時間點的價格。

Output

於同一行輸出兩個正整數，代表最大可能獲利以及最少需要幾次交易才能得到該獲利。

Scoring

- $2 \leq N \leq 10^6$
- $0 \leq C \leq 10^6$
- $0 \leq p_i \leq 10^6$

Subtask	Score	Constraints
1	11	$N \leq 20$
2	10	$N \leq 2000$, $C = 0$
3	19	$N \leq 2000$
4	47	$C = 0$
5	13	無額外限制

Examples

input
11 0 100 90 185 120 80 150 140 180 110 150 50
output
245 4

input
11 98 100 90 185 120 80 150 140 180 110 150 50
output
2 1

input
11 100 100 90 185 120 80 150 140 180 110 150 50
output
0 0

Note

範例測資二解釋： 以 80 買入， 180 賣出，得到利潤 $(180 - 100) - 98$ 元。

由於本題的輸入量十分龐大，如使用C++的cin讀入資料，請務必於main函式開頭加上
ios::sync_with_stdio(0),cin.tie(0); 以避免因輸入過慢而TLE。

蛋餅買蛋餅

時間限制: 2.5 秒

記憶體限制: 256 MiB

本題是一題互動題。

希望能成為 CP 大師的蛋餅，今天打完 SDVX 後，決定去買蛋餅吃。

看著架上滿滿的蛋餅，蛋餅不知如何做選擇，但他神祕的第六感告訴他，可能有一份蛋餅的重量與其他蛋餅不同!

店內有 N 份外觀相同的蛋餅，依序由 0 到 $N-1$ 號編號。其中 $N-1$ 份蛋餅皆為 100 公克，剩下一份蛋餅的重量可能為 90、100、110 公克。

店內只有一個無刻度的天秤，你可以選擇一些蛋餅放在天秤的左側，再選擇另一些蛋餅放在天秤的右側，以得知天秤哪一側蛋餅的總重較大，或兩側蛋餅總重相同。

當然，一個蛋餅不可能同時出現在天秤兩側，你也不能把**蛋餅本人**放上天秤，否則天秤會無法承受。

你的目標是使用盡量少次的測量找出重量不同的蛋餅編號為何，或判斷所有蛋餅的重量皆相同。

Interaction

無論 C/C++，請一律使用 C++14 上傳。

請注意: 請勿任意進行輸入/輸出。任何輸入/輸出可能造成評分結果為 0 分。

本題競賽頁面中有兩個檔案 "grader.h" 及 "example.cpp"。 請在程式開頭加上 #include"grader.h"，測試時請將範例檔案置於程式檔同一個資料夾。

互動時，有 4 個函數可以使用:

- **int Init()**

  於每筆測資開始時呼叫此函數，此函數將回傳該筆測資的蛋餅數量 N。

  如讀入 $N=0$，代表本組測資結束，請呼叫 **End**

  如在呼叫此函式前即呼叫其他函式，你將會得到 Wrong Answer: Please call Init first!

- **int measure(vector<int> L, vector<int> R)**

  將 L 內的蛋餅放置在天秤左端，R 內的蛋餅放置在右端。

  回傳值若為 -1 ，代表左端較重，若為 1，代表右端較重，若為 0，代表兩端等重。

  如詢問不合法，你將會得到 Wrong Answer: Invalid measure.

  如單筆詢問次數超過 N ，你將會得到 Wrong Answer: You measured too many times!

- **void Answer(int x)**

  回答重量不同的蛋餅編號，如所有蛋餅重量皆相同，請回答-1。

  若答案錯誤，你將會得到 Wrong Answer: Your answer is wrong!。

  若答案正確，評分程式將不會有任何動作，請呼叫 **Init** 讀取下一筆測資。

- **void End()**

當讀入 $N = 0$ 時，呼叫此函式以結束整組測資。呼叫此函數後程式將自動停止。

如未回答完所有測資即呼叫，你將會得到 Wrong_answer: You end the program too early!

- **函數之使用可參考 example.cpp**

一組測資中有多筆測資，請確保你的程式能處理多筆測資的情形。

測試時，請手動輸入測資，輸入格式如下。

n_1 p_1 w_1

n_2 p_2 w_2

n_3 p_3 w_3

.

.

.

0 0 0

其中 n_i 代表第 i 筆測資的蛋餅數量，p_i 代表第 i 筆測資中重量不同的蛋餅編號，w_i 代表第 i 筆測資中重量不同的蛋餅重量。

如該筆測資中所有蛋餅等重，則 $p_i = w_i = -1$。

Scoring

- 一筆測資中 $3 \leq N \leq 200$
- 一組測資中 N 的總和 ≤ 20000

Subtask	Score	Constraints
1	5	$N = 3$
2	10	$N \leq 10$
3	35	其中 $N - 1$ 份蛋餅重量為 100 克,剩下一份蛋餅的重量為 90 克。
4	50	無特殊限制。

對於子題 1 和 2，如果你的程式被判為 Accepted，該組測資你得到的分數即為該子題的滿分，否則為 0 分。

對於子題 3 和 4，如果你的程式被判為 Accepted，該組測資你得到的分數如下計算，否則為 0 分。

設該子題的滿分為 p，對於每組測資，你的最大詢問次數為 q，該組測資你的得分為：

- 如果 $q \leq 6$，你的分數是 p。
- 如果 $6 < q \leq 10$，你的分數是 $p \times (26 - q) \div 20$。
- 如果 $10 < q \leq 20$，你的分數是 $p \times (30 - q) \div 25$。
- 如果 $20 < q \leq N$，你的分數是 $p \div 5$。
- 否則，你的分數是 0。

每個子題的分數是該子題各組測資的最低分數。

Example

input
3 1 90 3 -1 -1

```
10 6 90
200 199 110
56 12 110
200 0 90
0 0 0
```

output

如正確引入grader.h, 程式將自動輸出結果