

第 1 題 成績指標

問題描述

一次考試中，於所有及格學生中獲取最低分數者最為幸運，反之，於所有不及格同學中，獲取最高分數者，可以說是最為不幸，而此二種分數，可以視為成績指標。

請你設計一支程式，讀入全班成績(人數不固定)，請對所有分數進行排序，並分別找出不及格中最高分數，以及及格中最低分數。

當找不到最低及格分數，表示對於本次考試而言，這是一個不幸之班級，此時請你印出：「worst case」；反之，當找不到最高不及格分數時，請你印出「best case」。

註：假設及格分數為 60，每筆測資皆為 0~100 間整數，且筆數未定。

輸入格式

第一行輸入學生人數，第二行為各學生分數(0~100 間)，分數與分數之間以一個空白間格。每一筆測資的學生人數為 1~20 的整數。

輸出格式

每筆測資輸出三行。

第一行由小而大印出所有成績，兩數字之間以一個空白間格，最後一個數字後無空白；

第二行印出最高不及格分數，如果全數及格時，於此行印出 best case；

第三行印出最低及格分數，當全數不及格時，於此行印出 worst case。

範例一：輸入

```
10
0 11 22 33 55 66 77 99 88 44
```

範例一：正確輸出

```
0 11 22 33 44 55 66 77 88 99
55
66
```

(說明) 不及格分數最高為 55，及格分數最低為 66。

範例二：輸入

```
1
13
```

範例二：正確輸出

```
13
13
worst case
```

(說明) 由於找不到最低及格分，因此第三行須印出「worst case」。

範例三：輸入

2
73 65

範例三：正確輸出

65 73
best case
65

(說明) 由於找不到不及格分，因此第二行須印出「best case」。

評分說明

輸入包含若干筆測試資料，每一筆測試資料的執行時間限制(time limit)均為 2 秒，依正確通過測資筆數給分。

第 2 題 矩陣轉換

問題描述

矩陣是將一群元素整齊的排列成一個矩形，在矩陣中的橫排稱為列 (row)，直排稱為行 (column)，其中以 X_{ij} 來表示矩陣 X 中的第 i 列第 j 行的元素。如圖一中， $X_{32} = 6$ 。

我們可以對矩陣定義兩種操作如下：

翻轉：即第一列與最後一列交換、第二列與倒數第二列交換、...依此類推。

旋轉：將矩陣以順時針方向轉 90 度。

例如：矩陣 X 翻轉後可得到 Y ，將矩陣 Y 再旋轉後可得到 Z 。

X	Y	Z
1 4	3 6	1 2 3
2 5	2 5	4 5 6
3 6	1 4	

圖一

一個矩陣 A 可以經過一連串的旋轉與翻轉操作後，轉換成新矩陣 B 。如圖二中， A 經過翻轉與兩次旋轉後，可以得到 B 。給定矩陣 B 和一連串的操作，請算出原始的矩陣 A 。

A	翻轉	旋轉	旋轉	B
1 1	→	2 1	→	1 1
1 3		1 3		3 1
2 1		1 1		1 2

圖二

輸入格式

第一行有三個介於 1 與 10 之間的正整數 R, C, M 。接下來有 R 行(line)是矩陣 B 的內容，每一行(line)都包含 C 個正整數，其中的第 i 行第 j 個數字代表矩陣 B_{ij} 的值。在矩陣內容後的一行有 M 個整數，表示對矩陣 A 進行的操作。第 k 個整數 m_k 代表第 k 個操作，如果 $m_k = 0$ 則代表旋轉， $m_k = 1$ 代表翻轉。同一行的數字之間都是以一個空白間格，且矩陣內容為 0~9 的整數。

輸出格式

輸出包含兩個部分。第一個部分有一行，包含兩個正整數 R' 和 C' ，以一個空白

隔開，分別代表矩陣 A 的列數和行數。接下來有 R' 行，每一行都包含 C' 個正整數，且每一行的整數之間以一個空白隔開，其中第 i 行的第 j 個數字代表矩陣 A_{ij} 的值。每一行的最後一個數字後並無空白。

範例一：輸入

```
3 2 3
1 1
3 1
1 2
1 0 0
```

範例一：正確輸出

```
3 2
1 1
1 3
2 1
```

(說明)

如圖二所示

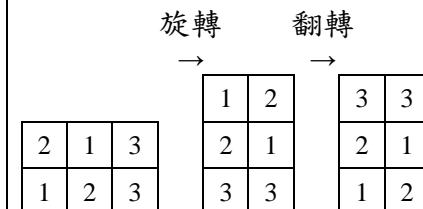
範例二：輸入

```
3 2 2
3 3
2 1
1 2
0 1
```

範例二：正確輸出

```
2 3
2 1 3
1 2 3
```

(說明)

**評分說明**

輸入包含若干筆測試資料，每一筆測試資料的執行時間限制(time limit)均為 2 秒，依正確通過測資筆數給分。其中：

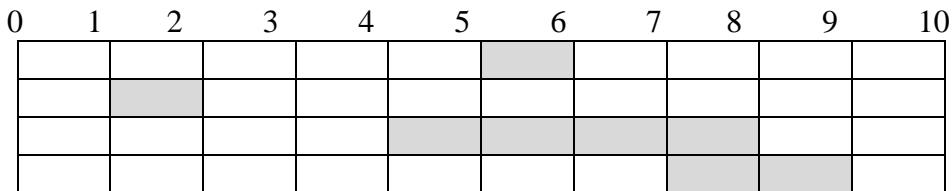
第一子題組共 30 分，其每個操作都是翻轉。

第二子題組共 70 分，操作有翻轉也有旋轉。

第 3 題 線段覆蓋長度

問題描述

給定一維座標上一些線段，求這些線段所覆蓋的長度，注意，重疊的部分只能算一次。例如給定三個線段：(5, 6)、(1, 2)、(4, 8)、和(7, 9)，如下圖，線段覆蓋長度為 6。



輸入格式：

第一列是一個正整數 N，表示此測試案例有 N 個線段。

接著的 N 列每一列是一個線段的開始端點座標和結束端點座標整數值，開始端點座標值小於等於結束端點座標值，兩者之間以一個空格區隔。

輸出格式：

輸出其總覆蓋的長度。

範例一：輸入

輸入	說明
5	此測試案例有 5 個線段
160 180	開始端點座標值與結束端點座標
150 200	開始端點座標值與結束端點座標
280 300	開始端點座標值與結束端點座標
300 330	開始端點座標值與結束端點座標
190 210	開始端點座標值與結束端點座標

範例一：輸出

輸出	說明
110	測試案例的結果

範例二：輸入

輸入	說明
1	此測試案例有 1 個線段
120 120	開始端點座標值與結束端點座標值

範例二：輸出

輸出	說明
0	測試案例的結果

評分說明

輸入包含若干筆測試資料，每一筆測試資料的執行時間限制(time limit)均為 2 秒，依正確通過測資筆數給分。每一個端點座標是一個介於 0~M 之間的整數，每筆測試案例線段個數上限為 N。其中：

第一子題組共 30 分， $M < 1000$ ， $N < 100$ ，線段沒有重疊。

第二子題組共 40 分， $M < 1000$ ， $N < 100$ ，線段可能重疊。

第三子題組共 30 分， $M < 100000000$ ， $N < 10000$ ，線段可能重疊。

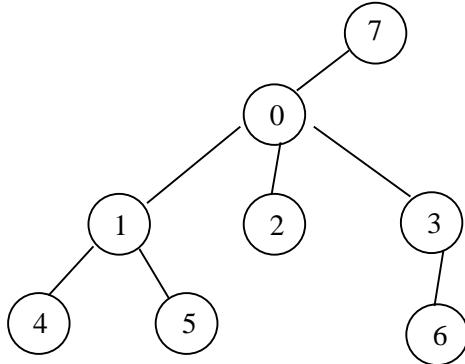
第 4 題 血緣關係

問題描述

小宇有一個大家族。有一天，他發現記錄整個家族成員和成員間血緣關係的家族族譜。小宇對於最遠的血緣關係（我們稱之為“血緣距離”）有多遠感到很好奇。

右圖為家族的關係圖。0 是 7 的孩子，1、2 和 3 是 0 的孩子，4 和 5 是 1 的孩子，6 是 3 的孩子。我們可以輕易的發現最遠的親戚關係為 4(或 5)和 6，他們的“血緣距離”是 4(4~1, 1~0, 0~3, 3~6)。

給予任一家族的關係圖，請找出最遠的“血緣距離”。你可以假設只有一個人是整個家族成員的祖先，而且沒有兩個成員有同樣的小孩。



輸入格式

第一行為一個正整數 n 代表成員的個數，每人以 0~n-1 之間惟一的編號代表。接著的 n-1 行，每行有兩個以一個空白隔開的整數 a 與 b ($0 \leq a, b \leq n-1$)，代表 b 是 a 的孩子。

輸出格式

每筆測資輸出一行最遠“血緣距離”的答案。

範例一：輸入

8
0 1
0 2
0 3
7 0
1 4
1 5
3 6

範例一：正確輸出

4

（說明）

如題目所附之圖，最遠路徑為 4->1->0->3->6 或 5->1->0->3->6，距離為 4。

範例二：輸入

4
0 1
0 2
2 3

範例二：正確輸出

3

（說明）

最遠路徑為 1->0->2->3，距離為 3。

評分說明

輸入包含若干筆測試資料，每一筆測試資料的執行時間限制(time limit)均為 3 秒，依正確通過測資筆數給分。其中，

第 1 子題組共 10 分，整個家族的祖先最多 2 個小孩，其他成員最多一個小孩， $2 \leq n \leq 100$ 。

第 2 子題組共 30 分， $2 \leq n \leq 100$ 。

第 3 子題組共 30 分， $101 \leq n \leq 2,000$ 。

第 4 子題組共 30 分， $1,001 \leq n \leq 100,000$ 。

第 1 題 三角形辨別

問題描述

當使用者輸入三角形之三邊長度，程式首先判別所輸入三邊常數值可否構成一個三角形，三角形之構成要素為任意兩邊相加皆大於第三邊，並於判別後分析該三角形屬於何種類型之三角形，三角形類別有：直角三角形、鈍角三角形、銳角三角形。請設計出一個能輸入三邊數值，輸出其是否為可構成三角形，倘若可構成為三角形，請輸出其所屬三角形類型。

提示：若 a 、 b 、 c 為三邊長，其中 c 為最長邊，則

$a + b \leq c$	無法構成三角形
$a \times a + b \times b < c \times c$	成立時為鈍角三角形(Obtuse triangle)
$a \times a + b \times b = c \times c$	成立時為直角三角形(Right triangle)
$a \times a + b \times b > c \times c$	成立時為銳角三角形(Acute triangle)

輸入格式

輸入為一行包含三邊長，邊長皆為小於 30001 的正整數，兩數之間有一空白。

輸出格式

每筆測資輸出兩行。

第一行由小而大印出三邊長，兩數字之間以一個空白間格，最後一個數字後無空白；

第二行輸出三角形的類型：

- 無法構成三角形輸出「No」；
- 鈍角三角形輸出「Obtuse」；
- 直角三角形輸出「Right」；
- 銳角三角形輸出「Acute」。

範例一：輸入

3 4 5

範例一：正確輸出

3 4 5

Right

(說明) $a \times a + b \times b = c \times c$ 成立時為直角三角形。

範例二：輸入

101 100 99

範例二：正確輸出

99 100 101

Acute

(說明) 邊長排序由小到大輸出， $a \times a + b \times b > c \times c$ 成立時為銳角三角形。

範例三：輸入

10 100 10

範例三：正確輸出

10 10 100

No

(說明) 由於無法構成三角形，因此第二行須印出「No」。

評分說明

輸入包含若干筆測試資料，每一筆測試資料的執行時間限制(time limit)均為 1 秒，依正確通過測資筆數給分。

第 2 題 最大和

問題描述

有 N 群數字，每群都恰有 M 個正整數，現在要求在每群中選擇一個數字，使得所有 N 群中選到的 N 個數字的總和是所有可能的選擇組合中最大的，計算出此總和 S 之後，輸出此總和 S ；此外，在那些被挑選的數字中，輸出可以整除 S 的數字，輸出順序則依照被選擇數字的群的先後順序。

輸入格式

第一行有二個正整數 N 和 M ， $1 \leq N \leq 20, 1 \leq M \leq 20$ 。

第二行開始有 N 行，每一行代表一群，一行中有 M 個正整數 x_i ，數字與數字間有一個空格，且其中 $1 \leq i \leq M$ ，以及 $1 \leq x_i \leq 256$ 。

輸出格式

第一行輸出最大總和 S 。

第二行按照被選擇數字所屬群的順序，輸出可以整除 S 的被選擇數字，數字與數字間以一個空格隔開，最後一個數字後無空白；若 N 個被選擇數字都不能整除 S ，就輸出-1。

範例一：輸入

3 2
1 5
6 4
1 1

範例一：正確輸出

12
6 1

(說明) 挑選的數字依序是 5,6,1，總和 $S=12$ 。而此三數中可整除 S 的是 6 與 1，6 在第二群，1 在第 3 群所以先輸出 6 再輸出 1。注意，1 雖然也出現在第一群，但她不是第一群中挑出的數字，所以順序是先 6 後 1。

範例二：輸入

4 3
6 3 2
2 7 9
4 7 1
9 5 3

範例二：正確輸出

31
-1

(說明) 挑選的數字依序是 6,9,7,9，總和 $S=31$ 。而此四數中沒有可整除 S 的，所以第二行輸出-1。

評分說明

輸入包含若干筆測試資料，每一筆測試資料的執行時間限制(time limit)均為 1 秒，依正確通過測資筆數給分。其中：

第 1 子題組 20 分： $1 \leq N \leq 20, M = 1$ 。

第 2 子題組 30 分： $1 \leq N \leq 20, M = 2$ 。

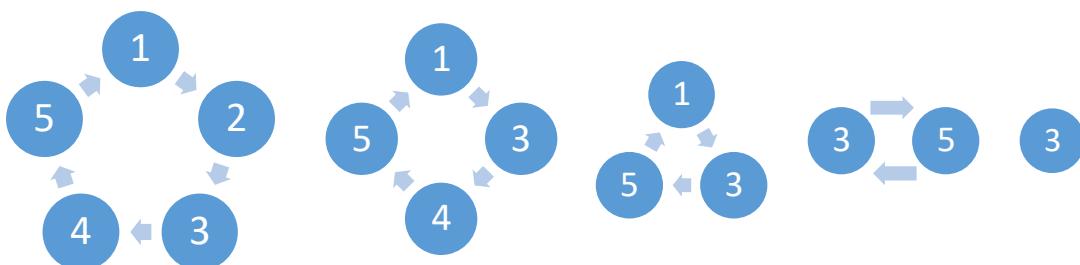
第 3 子題組 50 分： $1 \leq N \leq 20, 1 \leq M \leq 20$ 。

第 3 題 定時 K 彈

問題描述

「定時 K 彈」是一個團康遊戲， N 個人圍成一個圈，由 1 號依序到 N 號，從 1 號開始依序傳遞一枚玩具炸彈，炸彈每次到第 M 個人就會爆炸，此人即淘汰，被淘汰的人要離開圓圈，然後炸彈再從該淘汰者的下一個開始傳遞。遊戲之所以稱 K 彈是因為這枚炸彈只會爆炸 K 次，在第 K 次爆炸後，遊戲即停止，而此時在第 K 個淘汰者的下一位遊戲者被稱為幸運者，通常就會被要求表演節目。例如 $N=5, M=2$ ，如果 $K=2$ ，炸彈會爆炸兩次，被爆炸淘汰的順序依序是 2 與 4（參見下圖），這時 5 號就是幸運者。如果 $K=3$ ，剛才的遊戲會繼續，第三個淘汰的是 1 號，所以幸運者是 3 號。如果 $K=4$ ，下一輪淘汰 5 號，所以 3 號是幸運者。

此題輸入 N 、 M 與 K ，請你計算出誰是幸運者。



輸入格式

輸入只有一行包含三個正整數，依序為 N 、 M 與 K ，兩數中間有一個空格分開。其中 $1 \leq K < N$ 。

輸出格式

請輸出幸運者的號碼，結尾有換行符號。

範例一：輸入

5 2 4

範例一：正確輸出

3

（說明）

被淘汰的順序是 2、4、1、5，此時 5 的下一位是 3，也是最後剩下的，所以幸運者是 3。

範例二：輸入

8 3 6

範例二：正確輸出

4

（說明）

被淘汰的順序是 3、6、1、5、2、8，此時 8 的下一位是 4，所以幸運者是 4。

評分說明

輸入包含若干筆測試資料，每一筆測試資料的執行時間限制(time limit)均為 1 秒，依正確通過測資筆數給分。其中：

第 1 子題組 20 分， $1 \leq N \leq 100$ ，且 $1 \leq M \leq 10$ ， $K = N-1$ 。

第 2 子題組 30 分， $1 \leq N \leq 10,000$ ，且 $1 \leq M \leq 1,000,000$ ， $K = N-1$ 。

第 3 子題組 20 分， $1 \leq N \leq 200,000$ ，且 $1 \leq M \leq 1,000,000$ ， $K = N-1$ 。

第 4 子題組 30 分， $1 \leq N \leq 200,000$ ，且 $1 \leq M \leq 1,000,000$ ， $1 \leq K < N$ 。

第 4 題 棒球遊戲

問題描述

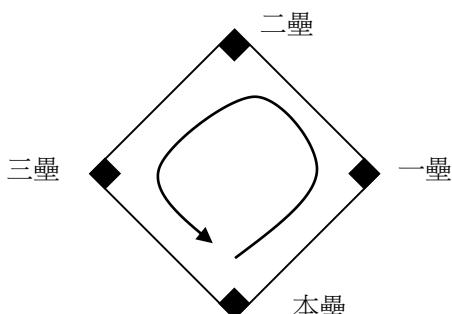
謙謙最近迷上棒球，他想自己寫一個簡化的棒球遊戲計分程式。這個程式會讀入球隊中每位球員的打擊結果，然後計算出球隊的得分。

這是個簡化版的模擬，假設擊球員的打擊結果只有以下情況：

- (1) 安打：以 1B, 2B, 3B 和 HR 分別代表一壘打、二壘打、三壘打和全（四）壘打。
- (2) 出局：以 FO, GO, 和 SO 表示。

這個簡化版的規則如下：

- (1) 球場上有四個壘包，稱為本壘、一壘、二壘和三壘。
- (2) 站在本壘握著球棒打球的稱為「擊球員」，站在另外三個壘包的稱為「跑壘員」。
- (3) 當擊球員的打擊結果為「安打」時，場上球員（擊球員與跑壘員）可以移動；結果為「出局」時，跑壘員不動，擊球員離場，換下一位擊球員。
- (4) 球隊總共有九位球員，依序排列。比賽開始由第 1 位開始打擊，當第 i 位球員打擊完畢後，由第 $(i+1)$ 位球員擔任擊球員。當第九位球員完畢後，則輪回第一位球員。
- (5) 當打出 K 壘打時，場上球員（擊球員和跑壘員）會前進 K 個壘包。從本壘前進一個壘包會移動到一壘，接著是二壘、三壘，最後回到本壘。
- (6) 每位球員回到本壘時可得 1 分。
- (7) 每達到三個出局數時，一、二和三壘就會清空（跑壘員都得離開），重新開始。



現在請你也寫出具備這樣功能的程式，計算球隊的總得分。

輸入格式

1. 每組測試資料固定有十行。
2. 第一到九行，依照球員順序，每一行代表一位球員的打擊資訊。每一行開始有一個正整數 a ($1 \leq a \leq 5$)，代表球員總共打了 a 次。接下來有 a 個字串（均為兩個字元），依序代表每次打擊的結果。資料之間均以一個空白字元隔開。球員的打擊資訊不會有錯誤也不會缺漏。
3. 第十行有一個正整數 b ($1 \leq b \leq 27$)，表示我們想要計算當總出局數累計到 b 時，該球隊的得分。輸入的打擊資訊中至少包含 b 個出局。

輸出格式

計算在總計第 b 個出局數發生時的總得分，並將此得分輸出於一行。

範例一：輸入

5 1B 1B FO GO 1B
5 1B 2B FO FO SO
4 SO HR SO 1B
4 FO FO FO HR
4 1B 1B 1B 1B
4 GO GO 3B GO
4 1B GO GO SO
4 SO GO 2B 2B
4 3B GO GO FO
3

範例一：正確輸出

0

(說明)

1B：一壘有跑壘員。
1B：一、二壘有跑壘員。
SO：一、二壘有跑壘員，一出局。
FO：一、二壘有跑壘員，兩出局。
1B：一、二、三壘有跑壘員，兩出局。
GO：一、二、三壘有跑壘員，三出局。

達到第三個出局數時，一、二、三壘均有跑壘員，但無法得分。因為 $b = 3$ ，代表三個出局就結束比賽，因此得到 0 分。

範例二：輸入

5 1B 1B FO GO 1B
5 1B 2B FO FO SO
4 SO HR SO 1B
4 FO FO FO HR
4 1B 1B 1B 1B
4 GO GO 3B GO
4 1B GO GO SO
4 SO GO 2B 2B
4 3B GO GO FO
6

範例二：正確輸出

5

(說明)接續範例一，達到第三個出局數時未得分，壘上清空。

1B：一壘有跑壘員。
SO：一壘有跑壘員，一出局。
3B：三壘有跑壘員，一出局，得一分。
1B：一壘有跑壘員，一出局，得兩分。
2B：二、三壘有跑壘員，一出局，得兩分。
HR：一出局，得五分。
FO：兩出局，得五分。
1B：一壘有跑壘員，兩出局，得五分。
GO：一壘有跑壘員，三出局，得五分。

因為 $b = 6$ ，代表我們要計算的是累積六個出局時的得分，因此在前 3 個出局數時得 0 分，第 4~6 個出局數得到 5 分，因此總得分是 $0+5=5$ 分。

評分說明

輸入包含若干筆測試資料，每一筆測試資料的執行時間限制(time limit)均為 1 秒，依正確通過測資筆數給分。其中：

- 第 1 子題組 20 分，打擊表現只有 HR 和 SO 兩種。
- 第 2 子題組 20 分，安打表現只有 1B，而且 b 固定為 3。
- 第 3 子題組 20 分， b 固定為 3。
- 第 4 子題組 40 分，無特別限制。

第 1 題 秘密差

問題描述

將一個十進位正整數的奇數位數的和稱為 A，偶數位數的和稱為 B，則 A 與 B 的絕對差值 $|A-B|$ 稱為這個正整數的秘密差。

例如：263541 的奇數位數的和 $A = 6+5+1 = 12$ ，偶數位數的和 $B = 2+3+4 = 9$ ，所以 263541 的秘密差是 $|12-9| = 3$ 。

給定一個十進位正整數 X，請找出 X 的秘密差。

輸入格式

輸入為一行含有一個十進位表示法的正整數 X，之後是一個換行字元。

輸出格式

請輸出 X 的秘密差 Y(以十進位表示法輸出)，以換行字元結尾。

範例一：輸入

263541

範例一：正確輸出

3

(說明) 263541 的 $A = 6+5+1 = 12$ ， $B = 2+3+4 = 9$ ， $|A-B| = |12-9| = 3$ 。

範例二：輸入

131

範例二：正確輸出

1

(說明) 131 的 $A = 1+1 = 2$ ， $B = 3$ ， $|A-B| = |2-3| = 1$ 。

評分說明

輸入包含若干筆測試資料，每一筆測試資料的執行時間限制(time limit)均為 1 秒，依正確通過測資筆數給分。其中：

第 1 子題組 20 分：X 一定恰好四位數。

第 2 子題組 30 分：X 的位數不超過 9。

第 3 子題組 50 分：X 的位數不超過 1000。

第 2 題 小群體

問題描述

Q 同學正在學習程式，P 老師出了以下的題目讓他練習。

一群人在一起時經常會形成一個一個的小群體。假設有 N 個人，編號由 0 到 N-1，每個人都寫下他最好朋友的編號（最好朋友有可能是他自己的編號，如果他自己沒有其他好友），在本題中，每個人的好友編號絕對不會重複，也就是說 0 到 N-1 每個數字都恰好出現一次。

這種好友的關係會形成一些小群體。例如 N=10，好友編號如下，

	0	1	2	3	4	5	6	7	8	9
好友編號	4	7	2	9	6	0	8	1	5	3

0 的好友是 4，4 的好友是 6，6 的好友是 8，8 的好友是 5，5 的好友是 0，所以 0、4、6、8、和 5 就形成了一個小群體。另外，1 的好友是 7 而且 7 的好友是 1，所以 1 和 7 形成另一個小群體，同理，3 和 9 是一個小群體，而 2 的好友是自己，因此他自己是一個小群體。總而言之，在這個例子裡有 4 個小群體： $\{0,4,6,8,5\}$ 、 $\{1,7\}$ 、 $\{3,9\}$ 、 $\{2\}$ 。本題的問題是：輸入每個人的好友編號，計算出總共有幾個小群體。

Q 同學想了想卻不知如何下手，和藹可親的 P 老師於是給了他以下的提示：如果你從任何一人 x 開始，追蹤他的好友，好友的好友，...，這樣一直下去，一定會形成一個圈回到 x，這就是一個小群體。如果我們追蹤的過程中把追蹤過的加以標記，很容易知道哪些人已經追蹤過，因此，當一個小群體找到之後，我們再從任何一個還未追蹤過的開始繼續找下一個小群體，直到所有的人都追蹤完畢。

Q 同學聽完之後很順利的完成了作業。

在本題中，你的任務與 Q 同學一樣：給定一群人的好友，請計算出小群體個數。

輸入格式

第一行是一個正整數 N，說明團體中人數。

第二行依序是 0 的好友編號、1 的好友編號、...、N-1 的好友編號。共有 N 個數字，包含 0 到 N-1 的每個數字恰好出現一次，數字間會有一個空白隔開。

輸出格式

請輸出小群體的個數。不要有任何多餘的字或空白，並以換行字元結尾。

範例一：輸入

10
4 7 2 9 6 0 8 1 5 3

範例二：輸入

3
0 2 1

範例一：正確輸出

4

（說明）

4 個小群體是 $\{0,4,6,8,5\}$, $\{1,7\}$, $\{3,9\}$ 和 $\{2\}$ 。

範例二：正確輸出

2

（說明）

2 個小群體分別是 $\{0\}$, $\{1,2\}$ 。

評分說明

輸入包含若干筆測試資料，每一筆測試資料的執行時間限制(time limit)均為 1 秒，依正確通過測資筆數給分。其中：

第 1 子題組 20 分， $1 \leq N \leq 100$ ，每一個小群體不超過 2 人。

第 2 子題組 30 分， $1 \leq N \leq 1,000$ ，無其他限制。

第 3 子題組 50 分， $1,001 \leq N \leq 50,000$ ，無其他限制。

第 3 題 數字龍捲風

問題描述

給定一個 $N \times N$ 的二維陣列，其中 N 是奇數，我們可以從正中間的位置開始，以順時針旋轉的方式走訪每個陣列元素恰好一次。對於給定的陣列內容與起始方向，請輸出走訪順序之內容。下面的例子顯示了 $N=5$ 且第一步往左的走訪順序：

3 _o	4 _o	2 _o	1 _o	4 _o
4 _o	2 _o	3 _o	8 _o	9 _o
2 _o	1 _o	9 _o	5 _o	6 _o
4 _o	2 _o	3 _o	7 _o	8 _o
1 _o	2 _o	6 _o	4 _o	3 _o

依此順序輸出陣列內容則可以得到「9123857324243421496834621」。

類似地，如果是第一步向上，則走訪順序如下：

3 _o	4 _o	2 _o	1 _o	4 _o
4 _o	2 _o	3 _o	8 _o	9 _o
2 _o	1 _o	9 _o	5 _o	6 _o
4 _o	2 _o	3 _o	7 _o	8 _o
1 _o	2 _o	6 _o	4 _o	3 _o

依此順序輸出陣列內容則可以得到「9385732124214968346214243」。

輸入格式

輸入第一行是整數 N , N 為奇數且不小於 3。第二行是一個 0~3 的整數代表起始方向，其中 0 代表左、1 代表上、2 代表右、3 代表下。第三行開始 N 行是陣列內容，順序是由上而下，由左至右，陣列的內容為 0~9 的整數，同一行數字中間以一個空白間隔。

輸出格式

請輸出走訪順序的陣列內容，該答案會是一連串的數字，數字之間不要輸出空白，結尾有換行符號。

範例一：輸入

5
0
3 4 2 1 4
4 2 3 8 9
2 1 9 5 6
4 2 3 7 8
1 2 6 4 3

範例一：正確輸出

9123857324243421496834621

範例二：輸入

3
1
4 1 2
3 0 5
6 7 8

範例二：正確輸出

012587634

評分說明

輸入包含若干筆測試資料，每一筆測試資料的執行時間限制(time limit)均為 1 秒，依正確通過測資筆數給分。其中：

第 1 子題組 20 分， $3 \leq N \leq 5$ ，且起始方向均為向左。

第 2 子題組 80 分， $3 \leq N \leq 49$ ，起始方向無限定。

提示：本題有多種處理方式，其中之一是觀察每次轉向與走的步數。例如，起始方向是向左時，前幾步的走法是：左 1、上 1、右 2、下 2、左 3、上 3、……一直到出界為止。

第 4 題 基地台

問題描述

為因應資訊化與數位化的發展趨勢，某市長想要在城市的一些服務點上提供無線網路服務，因此他委託電信公司架設無線基地台。某電信公司負責其中 N 個服務點，這 N 個服務點位在一條筆直的大道上，它們的位置(座標)係以與該大道一端的距離 $P[i]$ 來表示，其中 $i=0 \sim N-1$ 。由於設備訂製與維護的因素，每個基地台的服務範圍必須都一樣，當基地台架設後，與此基地台距離不超過 R (稱為基地台的半徑)的服務點都可以使用無線網路服務，也就是說每一個基地台可以服務的範圍是 $D=2R$ (稱為基地台的直徑)。現在電信公司想要計算，如果要架設 K 個基地台，那麼基地台的最小直徑是多少才能使每個服務點都可以得到服務。

基地台架設的地點不一定要在服務點上，最佳的架設地點也不唯一，但本題只需要求最小直徑即可。以下是一個 $N=5$ 的例子，五個服務點的座標分別是 1、2、5、7、8。



假設 $K=1$ ，最小的直徑是 7，基地台架設在座標 4.5 的位置，所有點與基地台的距離都在半徑 3.5 以內。假設 $K=2$ ，最小的直徑是 3，一個基地台服務座標 1 與 2 的點，另一個基地台服務另外三點。在 $K=3$ 時，直徑只要 1 就足夠了。

輸入格式

輸入有兩行。第一行是兩個正整數 N 與 K ，以一個空白間格。第二行 N 個非負整數 $P[0], P[1], \dots, P[N-1]$ 表示 N 個服務點的位置，這些位置彼此之間以一個空白間格。請注意，這 N 個位置並不保證相異也未經過排序。本題中， $K < N$ 且所有座標是整數，因此，所求最小直徑必然是不小於 1 的整數。

輸出格式

輸出最小直徑，不要有任何多餘的字或空白並以換行結尾。

範例一：輸入

5 2
5 1 2 8 7

範例一：正確輸出

3
(說明) 如題目中之說明。

範例二：輸入

5 1
7 5 1 2 8

範例二：正確輸出

7
(說明) 如題目中之說明。

評分說明

輸入包含若干筆測試資料，每一筆測試資料的執行時間限制(time limit)均為 2 秒，依正確通過測資筆數給分。其中：

第 1 子題組 10 分，座標範圍不超過 100 ， $1 \leq K \leq 2$ ， $K < N \leq 10$ 。

第 2 子題組 20 分，座標範圍不超過 $1,000$ ， $1 \leq K < N \leq 100$ 。

第 3 子題組 20 分，座標範圍不超過 $1,000,000,000$ ， $1 \leq K < N \leq 500$ 。

第 4 子題組 50 分，座標範圍不超過 $1,000,000,000$ ， $1 \leq K < N \leq 50,000$ 。

邏輯運算子 (Logic Operators)

問題描述

小蘇最近在學三種邏輯運算子 AND、OR 和 XOR。這三種運算子都是二元運算子，也就是說在運算時需要兩個運算元，例如 $a \text{ AND } b$ 。對於整數 a 與 b ，以下三個二元運算子的運算結果定義如下列三個表格：

a AND b		
	b 為 0	b 不為 0
a 為 0	0	0
a 不為 0	0	1

a OR b		
	b 為 0	b 不為 0
a 為 0	0	1
a 不為 0	1	1

a XOR b		
	b 為 0	b 不為 0
a 為 0	0	1
a 不為 0	1	0

舉例來說：

- (1) $0 \text{ AND } 0$ 的結果為 0， $0 \text{ OR } 0$ 以及 $0 \text{ XOR } 0$ 的結果也為 0。
- (2) $0 \text{ AND } 3$ 的結果為 0， $0 \text{ OR } 3$ 以及 $0 \text{ XOR } 3$ 的結果則為 1。
- (3) $4 \text{ AND } 9$ 的結果為 1， $4 \text{ OR } 9$ 的結果也為 1，但 $4 \text{ XOR } 9$ 的結果為 0。

請撰寫一個程式，讀入 a 、 b 以及邏輯運算的結果，輸出可能的邏輯運算為何。

輸入格式

輸入只有一行，共三個整數值，整數間以一個空白隔開。第一個整數代表 a ，第二個整數代表 b ，這兩數均為非負的整數。第三個整數代表邏輯運算的結果，只會是 0 或 1。

輸出格式

輸出可能得到指定結果的運算，若有多個，輸出順序為 AND、OR、XOR，每個可能的運算單獨輸出一行，每行結尾皆有換行。若不可能得到指定結果，輸出 IMPOSSIBLE。(注意輸出時所有英文字母均為大寫字母。)

範例一：輸入

0 0 0

範例一：正確輸出

AND

OR

XOR

範例三：輸入

3 0 1

範例三：正確輸出

OR

XOR

範例二：輸入

1 1 1

範例二：正確輸出

AND

OR

範例四：輸入

0 0 1

範例四：正確輸出

IMPOSSIBLE

評分說明

輸入包含若干筆測試資料，每一筆測試資料的執行時間限制(time limit)均為 1 秒，依正確通過測資筆數給分。其中：

第 1 子題組 80 分， a 和 b 的值只會是 0 或 1。

第 2 子題組 20 分， $0 \leq a, b < 10,000$ 。

交錯字串 (Alternating Strings)

問題描述

一個字串如果全由大寫英文字母組成，我們稱為大寫字串；如果全由小寫字母組成則稱為小寫字串。字串的長度是它所包含字母的個數，在本題中，字串均由大小寫英文字母組成。假設 k 是一個自然數，一個字串被稱為「 k -交錯字串」，如果它是由長度為 k 的大寫字串與長度為 k 的小寫字串交錯串接組成。

舉例來說，「StRiNg」是一個 1-交錯字串，因為它是一個大寫一個小寫交替出現；而「heLLow」是一個 2-交錯字串，因為它是兩個小寫接兩個大寫再接兩個小寫。但不管 k 是多少，「aBBaaa」、「BaBaBB」、「aaaAAbbCCCC」都不是 k -交錯字串。

本題的目標是對於給定 k 值，在一個輸入字串找出最長一段連續子字串滿足 k -交錯字串的要求。例如 $k=2$ 且輸入「aBBaaa」，最長的 k -交錯字串是「Baa」，長度為 4。又如 $k=1$ 且輸入「BaBaBB」，最長的 k -交錯字串是「BaBaB」，長度為 5。

請注意，滿足條件的子字串可能只包含一段小寫或大寫字母而無交替，如範例二。此外，也可能不存在滿足條件的子字串，如範例四。

輸入格式

輸入的第一行是 k ，第二行是輸入字串，字串長度至少為 1，只由大小寫英文字母組成(A~Z, a~z)並且沒有空白。

輸出格式

輸出輸入字串中滿足 k -交錯字串的要求的最長一段連續子字串的長度，以換行結尾。

範例一：輸入

1
aBBdaaa

範例一：正確輸出

2

範例三：輸入

2
aafAXbbCDCCC

範例三：正確輸出

8

範例二：輸入

3
DDaasAAAbbCC

範例二：正確輸出

3

範例四：輸入

3
DDaaAAbbCC

範例四：正確輸出

0

評分說明：

輸入包含若干筆測試資料，每一筆測試資料的執行時間限制(time limit)均為 1 秒，依正確通過測資筆數給分。其中：

第 1 子題組 20 分，字串長度不超過 20 且 $k=1$ 。

第 2 子題組 30 分，字串長度不超過 100 且 $k \leq 2$ 。

第 3 子題組 50 分，字串長度不超過 100,000 且無其他限制。

提示：根據定義，要找的答案是大寫片段與小寫片段交錯串接而成。本題有多種解法的思考方式，其中一種是從左往右掃描輸入字串，我們需要紀錄的狀態包含：目前是在小寫子

字串中還是大寫子字串中，以及在目前大(小)寫子字串的第幾個位置。根據下一個字母的大小寫，我們需要更新狀態並且記錄以此位置為結尾的最長交替字串長度。

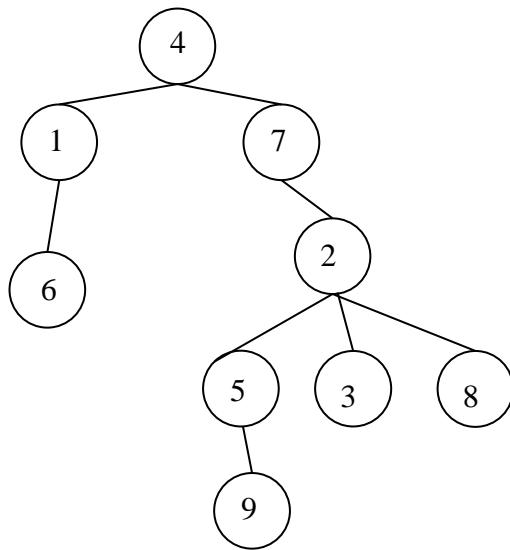
另外一種思考是先掃描一遍字串，找出每一個連續大(小)寫片段的長度並將其記錄在一個陣列，然後針對這個陣列來找出答案。

樹狀圖分析 (Tree Analyses)

問題描述

本題是關於有根樹(rooted tree)。在一棵 n 個節點的有根樹中，每個節點都是以 1~n 的不同數字來編號，描述一棵有根樹必須定義節點與節點之間的親子關係。一棵有根樹恰有一個節點沒有父節點(parent)，此節點被稱為根節點(root)，除了根節點以外的每一個節點都恰有一個父節點，而每個節點被稱為是它父節點的子節點(child)，有些節點沒有子節點，這些節點稱為葉節點(leaf)。在當有根樹只有一個節點時，這個節點既是根節點同時也是葉節點。

在圖形表示上，我們將父節點畫在子節點之上，中間畫一條邊(edge)連結。例如，圖一中表示的是一棵 9 個節點的有根樹，其中，節點 1 為節點 6 的父節點，而節點 6 為節點 1 的子節點；又 5、3 與 8 都是 2 的子節點。節點 4 沒有父節點，所以節點 4 是根節點；而 6、9、3 與 8 都是葉節點。



圖一

樹狀圖中的兩個節點 u 和 v 之間的距離 $d(u,v)$ 定義為兩節點之間邊的數量。如圖一中， $d(7, 5) = 2$ ，而 $d(1, 2) = 3$ 。對於樹狀圖中的節點 v ，我們以 $h(v)$ 代表節點 v 的高度，其定義是節點 v 和節點 v 下面最遠的葉節點之間的距離，而葉節點的高度定義為 0。如圖一中，節點 6 的高度為 0，節點 2 的高度為 2，而節點 4 的高度為 4。此外，我們定義 $H(T)$ 為 T 中所有節點的高度總和，也就是說 $H(T) = \sum_{v \in T} h(v)$ 。給定一個樹狀圖 T ，請找出 T 的根節點以及高度總和 $H(T)$ 。

輸入格式

第一行有一個正整數 n 代表樹狀圖的節點個數，節點的編號為 1 到 n 。接下來有 n 行，第 i 行的第一個數字 k 代表節點 i 有 k 個子節點，第 i 行接下來的 k 個數字就是這些子節點的編號。每一行的相鄰數字間以空白隔開。

輸出格式

輸出兩行各含一個整數，第一行是根節點的編號，第二行是 $H(T)$ 。

範例一：輸入

```
7  
0  
2 6 7  
2 1 4  
0  
2 3 2  
0  
0
```

範例一：正確輸出

```
5  
4
```

範例二：輸入

```
9  
1 6  
3 5 3 8  
0  
2 1 7  
1 9  
0  
1 2  
0  
0
```

範例二：正確輸出

```
4  
11
```

評分說明：

輸入包含若干筆測試資料，每一筆測試資料的執行時間限制(time limit)均為 1 秒，依正確通過測資筆數給分。測資範圍如下，其中 k 是每個節點的子節點數量上限：

第 1 子題組 10 分， $1 \leq n \leq 4, k \leq 3$ ，除了根節點之外都是葉節點。

第 2 子題組 30 分， $1 \leq n \leq 1,000, k \leq 3$ 。

第 3 子題組 30 分， $1 \leq n \leq 100,000, k \leq 3$ 。

第 4 子題組 30 分， $1 \leq n \leq 100,000, k$ 無限制。

提示：輸入的資料是給每個節點的子節點有哪些或沒有子節點，因此，可以根據定義找出根節點。關於節點高度的計算，我們根據定義可以找出以下遞迴關係式：(1)葉節點的高度為 0；(2)如果 v 不是葉節點，則 v 的高度是它所有子節點的最大高度加一。也就是說，假設 v 的子節點有 a, b 與 c ，則 $h(v) = \max\{ h(a), h(b), h(c) \} + 1$ 。以遞迴方式可以計算出所有節點的高度。

物品堆疊(Stacking)

問題描述

某個自動化系統中有一個存取物品的子系統，該系統是將 N 個物品堆在一個垂直的貨架上，每個物品各佔一層。系統運作的方式如下：每次只會取用一個物品，取用時必須先將在其上方的物品貨架升高，取用後必須將該物品放回，然後將剛才升起的貨架降回原始位置，之後才會進行下一個物品的取用。

每一次升高某些物品所需要消耗的能量是以這些物品的總重來計算，在此我們忽略貨架的重量以及其他可能的消耗。現在有 N 個物品，第 i 個物品的重量是 $w(i)$ 而需要取用的次數為 $f(i)$ ，我們需要決定如何擺放這些物品的順序來讓消耗的能量越小越好。舉例來說，有兩個物品 $w(1)=1$ 、 $w(2)=2$ 、 $f(1)=3$ 、 $f(2)=4$ ，也就是說物品 1 的重量是 1 需取用 3 次，物品 2 的重量是 2 需取用 4 次。我們有兩個可能的擺放順序(由上而下)：

- (1,2)，也就是物品 1 放在上方，2 在下方。那麼，取用 1 的時候不需要能量，而每次取用 2 的能量消耗是 $w(1)=1$ ，因為 2 需取用 $f(2)=4$ 次，所以消耗能量數為 $w(1)*f(2)=4$ 。
- (2,1)，也就是物品 2 放在 1 的上方。那麼，取用 2 的時候不需要能量，而每次取用 1 的能量消耗是 $w(2)=2$ ，因為 1 需取用 $f(1)=3$ 次，所以消耗能量數= $w(2)*f(1)=6$ 。

在所有可能的兩種擺放順序中，最少的能量是 4，所以答案是 4。再舉一例，若有三物品而 $w(1)=3$ 、 $w(2)=4$ 、 $w(3)=5$ 、 $f(1)=1$ 、 $f(2)=2$ 、 $f(3)=3$ 。假設由上而下以(3,2,1)的順序，此時能量計算方式如下：取用物品 3 不需要能量，取用物品 2 消耗 $w(3)*f(2)=10$ ，取用物品 1 消耗 $(w(3)+w(2))*f(1)=9$ ，總計能量為 19。如果以(1,2,3)的順序，則消耗能量為 $3*2+(3+4)*3=27$ 。事實上，我們一共有 $3!=6$ 種可能的擺放順序，其中順序(3,2,1)可以得到最小消耗能量 19。

輸入格式

輸入的第一行是物品件數 N ，第二行有 N 個正整數，依序是各物品的重量 $w(1)$ 、 $w(2)$ 、...、 $w(N)$ ，重量皆不超過 1000 且以一個空白間隔。第三行有 N 個正整數，依序是各物品的取用次數 $f(1)$ 、 $f(2)$ 、...、 $f(N)$ ，次數皆為 1000 以內的正整數，以一個空白間隔。

輸出格式

輸出最小能量消耗值，以換行結尾。所求答案不會超過 63 個位元所能表示的正整數。

範例一(第 1、3 子題)：輸入

2
20 10
1 1

範例一：正確輸出

10

範例二(第 2、4 子題)：輸入

3
3 4 5
1 2 3

範例二：正確輸出

19

評分說明：輸入包含若干筆測試資料，每一筆測試資料的執行時間限制(time limit)均為 1 秒，依正確通過測資筆數給分。其中：

第 1 子題組 10 分， $N = 2$ ，且取用次數 $f(1)=f(2)=1$ 。

第 2 子題組 20 分， $N = 3$ 。

第 3 子題組 45 分， $N \leq 1,000$ ，且每一個物品 i 的取用次數 $f(i)=1$ 。

第 4 子題組 25 分， $N \leq 100,000$ 。

第 1 題 追蹤目標

問題描述

調查局最近正在追查一個神祕案件，發現到蘇菲女士是重要關係人，為了保護蘇菲女士，調查局在她身上裝了追蹤器以掌握她的位置。為了不讓其他人竊聽獲知她的位置，追蹤器傳回的訊號會加入許多雜訊，而確切資訊會隱藏在一堆整數值裡，當追蹤器停止時會傳回整數 0。追蹤器有一個參數是正整數 K，解讀方式如下：

1. 一個整數的「最高 K 位數」是指擷取最左邊 K 位數所形成的數；而「最低 K 位數」是指擷取最右邊 K 位數所形成的數。例如，若 $K=2$ ，則 12345 的最高 K 位數是 12，最低 K 位數是 45；若 $K=3$ ，則 12345 的最高 K 位數是 123，最低 K 位是 345。
2. 蘇菲女士沿 X 軸的移動距離是所有整數的最高 K 位數的最大值，而沿 Y 軸的移動距離則是所有整數的最低 K 位數的最小值。

舉例而言，假設 $K=1$ ，如果追蹤器傳回的訊號為 354、9158 和 62，這代表蘇菲女士沿著 X 軸移動了 $\max\{3, 9, 6\} = 9$ 單位，沿著 Y 軸移動了 $\min\{4, 8, 2\} = 2$ 單位。假設 $K=2$ ，如果追蹤器傳回的訊號依舊為 354、9158 和 62，這代表沿著 X 軸移動了 $\max\{35, 91, 62\} = 91$ 單位，沿著 Y 軸移動了 $\min\{54, 58, 62\} = 54$ 單位。現在請你撰寫一個程式，讀入追蹤器傳回的訊號，決定蘇菲女士現在的位置。假設蘇菲女士一開始的位置必定為 $(0, 0)$ 。

輸入格式

1. 第一行有一個正整數 K ($1 \leq K \leq 9$)，代表追蹤器的參數。
2. 第二行有不超過 10 個非負整數，數字間以空白隔開，最後一個數必為 0 且此數不納入計算移動距離。除了最後的 0 之外，每個整數的位數至少為 K 位，數值小於 10^9 。

輸出格式

輸出兩個整數，代表蘇菲女士最後位置的 X 與 Y 座標，兩個整數間以一個空白隔開。

範例一：輸入

1
123 0

範例一：正確輸出

1 3

範例二：輸入

1
354 9158 62 0

範例二：正確輸出

9 2

範例三：輸入

2
678 0

範例三：正確輸出

67 78

範例四：輸入

2
354 9158 62 0

範例四：正確輸出

91 54

評分說明

輸入包含若干筆測試資料，每一筆測試資料的執行時間限制(time limit)均為 1 秒，依正確通過測資筆數給分。其中：

第 1 子題組 20 分：K=1，第二行除了結尾 0 之外只會有一個整數值，且必為兩位數。

第 2 子題組 40 分：K=1，第二行除了結尾 0 之外的整數必為兩位數。

第 3 子題組 40 分：無其他限制。

第 2 題 重複字串

問題描述

一個字串中如果有某些片段連續重複的出現，我們可以利用描述重複次數的編碼方式來表示這個字串，在某些適當的狀況下可以壓縮字串長度。例如我們用 3A2B 來表示 AAABB，其中 3A2B 稱為壓縮字串，AAABB 稱為解壓縮後字串。更精確來說，這樣編碼過的字串是一連串的數字與文字交錯出現的形式，文字稱為「樣式」，數字稱為「重複次數」，在每一段文字「樣式」之前的數字表示該「樣式」重複了多少次，「重複次數」一定是一個正整數。除了上面 3A2B 的例子外，我們也可以用 3CA4D 來表示 CACACADDDD，也就是說樣式的長度不一定是 1。本題的任務是將輸入的壓縮字串轉換為解壓縮後字串。

輸入格式

輸入是一個壓縮字串。字串長度不超過 100。這個字串完全由數字與大寫英文字母組成，沒有空白與任何其他字元。

輸出格式

輸出解壓縮後字串。解壓縮後長度不超過 1000。

範例一：輸入

3A

範例一：正確輸出

AAA

(說明：A 重複 3 次)

範例二：輸入

2CX5D

範例二：正確輸出

CXCXDDDD

(說明：CX 重複 2 次，然後 D 重複 5 次)

範例三：輸入

1A1P1C1S

範例四：輸入

1AB12F

範例三：正確輸出

APCS

範例四：正確輸出

ABFFFFFF

評分說明

輸入包含若干筆測試資料，每一筆測試資料的執行時間限制(time limit)均為 1 秒，依正確通過測資筆數給分。其中：

第 1 子題組 20 分：輸入的壓縮字串長度必然是 2，也就是說只有一組「重複次數」與「樣式」，而重複次數是一位數，樣式也只有一個字母(類似範例一的情形)。

第2子題組30分：每一個「重複次數」都只有一位整數(類似範例二與三的情形)。

第3子題組50分：無其他限制。

第 3 題 公平分區

問題描述

甲國是一個國土狹長的國家，甲國的 N 個城鎮都沿著一條筆直的大道均勻分布。我們可以把這些城鎮看成在一條直線上編號 $1, 2, \dots, N$ 的點，而相鄰兩點的距離都是 1。每個城鎮擁有若干人口，第 i 個城鎮的人口數以 $P[i]$ 表示，另外我們用 $[i, j]$ 表示包含從 i 到 j 這些城鎮所形成的連續區域，其中 $i \leq j$ 。

因為國土狹長，甲國人希望能夠將國土劃分層級以便分區治理，但是也不希望劃分的層級太多，因此有個層級的上限 K 。一個行政區是某個連續區域 $[i, j]$ ，全國是一個第 0 級行政區 $[1, N]$ 。行政區的劃分原則如下：

對於第 H 級行政區 $[S, T]$ ，如果滿足 $H < K$ 且至少包含三個城鎮 ($T-S \geq 2$)，則此行政區中就會選出一個城鎮 P ， $S < P < T$ ，在城鎮 P 設置行政機構並將這行政區劃分成左右兩個 $(H+1)$ 級的非空行政區域 $[S, P-1]$ 與 $[P+1, T]$ 。城鎮 P 的選擇方式是必須使得 $[S, P-1]$ 與 $[P+1, T]$ 左右兩個行政區人民到達 P 城鎮的總距離差異越小越好，如果有兩個城鎮可以選擇，則會選擇編號較小的城鎮。

舉例來說，甲國的各城鎮人口如下表：

編號	1	2	3	4	5	6	7
人口數	2	4	1	3	7	6	9

全國 $[1, 7]$ 是一個行政區，我們可以找到 $P=5$ ，左邊區域 $[1, 4]$ 的人民到達 P 的距離總和是： $2 \times (5 - 1) + 4 \times (5 - 2) + 1 \times (5 - 3) + 3 \times (5 - 4) = 25$ ，而右邊區域 $[6, 7]$ 的人民到達 P 距離總和是： $6 \times (6 - 5) + 9 \times (7 - 5) = 24$ 。總距離差異為 $|25 - 24| = 1$ ，這個差值是所有可能選擇 P 城鎮中最小的，所以甲國人會在 $P=5$ 設置行政機構將全國劃分成 $[1, 4]$ 與 $[6, 7]$ 兩個 1 級行政區。假設 $K=1$ ，則兩個 1 級行政區都不再考慮劃分，編號 5 的城鎮是唯一設置行政機構的城鎮。

現在我們假設人口數如上述但 $K=3$ 。那麼就必須再考慮那兩個 1 級行政區是否可進一步劃分。其中 $[6, 7]$ 不能再劃分了，因為至少要包含 3 個城鎮才能劃分。但是左邊的 $[1, 4]$ 可以進一步劃分，若以 $P=2$ 劃分成 $[1, 1]$ 與 $[3, 4]$ 兩區，兩邊的距離總和分別是： $2 \times (2 - 1) = 2$ 以及 $1 \times (3 - 2) + 3 \times (4 - 2) = 7$ ，差異是 5；而若以 $P=3$ 劃分成 $[1, 2]$ 與 $[4, 4]$ ，則兩邊的總距離差異剛好也是 5。兩處條件一樣，因此會選擇編號較小的 $P=2$ 來設置行政機構。我們也可看出，即使分層的上限 $K=3$ 還沒超過，但所劃分的 $[1, 1]$ 與 $[3, 4]$ 兩個 2 級行政區都不能再劃分了。所以這個例子中，設置行政機構的城鎮為 5 與 2。

給定甲國各個城鎮的人口數，請幫忙找出甲國人如何劃分行政區，計算出所有設置行政機構的城鎮人口總數。請注意，計算過程中，總距離可能超過 2^{31} 但不會超過 2^{60} 。

輸入格式

1. 第一行有兩個正整數 N 與 K，分別是甲國的城鎮數與分層上限。
2. 第二行有 N 個正整數，依序代表各個城鎮的人口數 $P[1] \sim P[N]$ ，數字間以空白隔開，人口數總和不超過 10^9 。

輸出格式

輸出所有設置行政機構的城鎮人口總數。

範例一：輸入

7 1
2 4 1 3 7 6 9

範例一：正確輸出

7
(說明)只有城鎮 5 設置了行政機構，它的人口數是 7

範例二：輸入

7 3
2 4 1 3 7 6 9

範例二：正確輸出

11
(說明)只有城鎮 5 與 2 設置了行政機構，人口總數是 $7+4=11$

評分說明

輸入包含若干筆測試資料，每一筆測試資料的執行時間限制(time limit)均為 1 秒，依正確通過測資筆數給分。其中：

第 1 子題組 20 分： $N \leq 100$ ，行政區劃分層級限制 $K=1$ ，也就是說，只要考慮一次劃分位置。

第 2 子題組 30 分： $N \leq 200$ ，行政區劃分層級限制 $K < 15$ 。

第 3 子題組 50 分： $N \leq 50000$ ，行政區劃分層級限制 $K < 30$ 。

第 4 題 階梯數字

問題敘述

阿明最近在學習程式語言，他對一些特別的正整數很有興趣，他發現一些十進數的每一位數已排好順序，從高位數往低位數看過去，每一位數字只會相等或變大，例如：9、234、777、11222233 等數字都有這性質，他稱這些數字為階梯數字。給定一正整數 N，阿明想知道不大於 N 的階梯數字總共有幾個，請注意本題只算正整數，所以 0 不算階梯數字，而且階梯數字不會以 0 開始。請幫阿明寫計算階梯數字的個數。

提示：對於 N 不大的情形，例如 100000，我們可以枚舉每一個不超過 N 的數字來判斷。對於 N 很大的時候，就需要比較快速的方法了。我們知道所有的一位數 1~9 都是階梯數字，如果要計算 D 位數的階梯數字總數，我們可以依據最高位數字分成 9 類，然後根據(D - 1)位的 9 類階梯數字總數來計算。

輸入格式

一個正整數 N。

輸出格式

輸出不大於 N 的階梯數字總個數。

範例一：輸入

25

範例一：正確輸出

22

(說明：1~9, 11~19, 22~25，共 22 個)

範例三：輸入

54321

範例三：正確輸出

1875

範例二：輸入

23456

範例二：正確輸出

1365

範例四：輸入

88888888

範例四：正確輸出

24301

評分說明

輸入包含若干筆測試資料，每一筆測試資料的執行時間限制(time limit)均為 1 秒，依正確通過測資筆數給分。其中：

第 1 子題組 30 分： $N \leq 10^5$ 。

第 2 子題組 20 分： $N = a \times 10^b$ ，a 與 b 均為整數，且 $1 \leq a \leq 9$ ， $10 \leq b \leq 17$ 。

第 3 子題組 50 分： $N \leq 10^{18}$ 。

第 1 題 特殊編碼

問題描述

任何文字與數字在電腦中儲存時都是使用二元編碼，而所謂二元編碼也就是一段由 0 與 1 構成的序列。在本題中，A~F 這六個字元由一種特殊方式來編碼，在這種編碼方式中，這六個字元的編碼都是一個長度為 4 的二元序列，對照表如下：

字元	A	B	C	D	E	F
編碼	0 1 0 1	0 1 1 1	0 0 1 0	1 1 0 1	1 0 0 0	1 1 0 0

請你寫一個程式從編碼辨識這六個字元。

輸入格式

第一行是一個正整數 N ， $1 \leq N \leq 4$ ，以下有 N 行，每行有 4 個 0 或 1 的數字，數字間彼此以空白隔開，每一行必定是上述六個字元其中之一的編碼。

輸出格式

輸出編碼所代表的 N 個字元，字元之間不需要空白或換行間格。

範例一：輸入

```
1
0 1 0 1
```

範例一：正確輸出

A

範例三：輸入

```
2
1 0 0 0
1 1 0 0
```

範例三：正確輸出

EF

範例二：輸入

```
1
0 0 1 0
```

範例二：正確輸出

C

範例四：輸入

```
4
1 1 0 1
1 0 0 0
0 1 1 1
1 1 0 1
```

範例四：正確輸出

DEBD

評分說明

輸入包含若干筆測試資料，每一筆測試資料的執行時間限制均為 1 秒，依正確通過測資筆數給分。其中：

第 1 子題組 50 分： $N = 1$ 。

第 2 子題組 50 分： $N \leq 4$ 。

第 2 題 完全奇數

問題描述

如果一個正整數的每一位數都是奇數時，例如：7、19、1759977 等，我們稱這種數字為完全奇數。對於輸入的一正整數 N ，如果 K 是最靠近 N 的完全奇數，請寫一程式找出 K 與 N 之間差距的絕對值，也就是說，請計算並輸出 $|K - N|$ 。

以 $N = 13256$ 為例，比 13256 大的最小完全奇數是 13311 ，比它小的最大完全奇數是 13199 ，因為 $|13311 - 13256| = 55 < |13256 - 13199| = 57$ ，因此輸出 55 。

輸入格式

一個正整數 N ， $N < 10^{18}$ 。

輸出格式

輸出 N 與其最近的完全奇數的差距。

範例一：輸入

135

範例一：正確輸出

0

範例三：輸入

35001

範例三：正確輸出

110

範例二：輸入

13256

範例二：正確輸出

55

範例四：輸入

1001

範例四：正確輸出

2

評分說明

輸入包含若干筆測試資料，每一筆測試資料的執行時間限制均為 1 秒，依正確通過測資筆數給分。其中：

第 1 子題組 20 分： $N < 100$ 。

第 2 子題組 30 分： $N < 10^6$ 。

第 3 子題組 50 分： $N < 10^{18}$ 。

第 3 題 工作排程

問題描述

有 M 個工作要在 N 台機器上加工，每個工作 i 包含若干個工序 o_{ij} ，這些工序必須依序加工，也就是前一道工序 $o_{i(j-1)}$ 完成後才可以開始下一道工序 o_{ij} 。每一道工序 o_{ij} 可以用一個有序對 (k_{ij}, t_{ij}) 來表示它需要在機器 k_{ij} 上面花費 t_{ij} 小時來完成。每台機器一次只能處理一道工序。

所謂一道工序 o_{ij} 的「最早完成時間 c_{ij}^* 」是指考慮目前排程中機器 k_{ij} 之可用性以及前一道工序 $o_{i(j-1)}$ (如果該工序存在) 之完成時間後可得的最早完成時間。工廠經理安排所有工序的排程規則如下：

針對每一個工作的第一個尚未排程的工序，計算出此工序的「最早完成時間」，然後挑選出最早完成時間最小的工序納入排程，如果有多個最早完成時間都是最小，則挑選其中工作編號最小之工序。一個工序一旦納入排程就不會再更改，重複以上步驟直到所有工序皆納入排程。

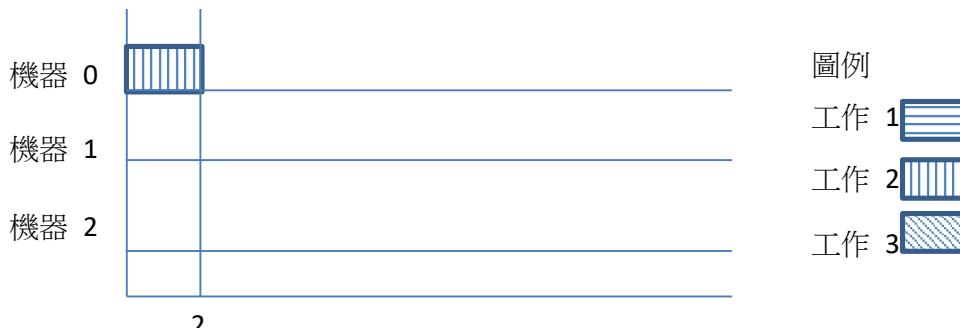
我們總是從時間 0 開始排程，每個工作的完成時間為其最後一個工序的完成時間，本題的目標是計算出每個工作的完成時間並輸出其總和。

以下以一個例子來說明，在此例中，有三個工作要在三台機器上排程，各工作的資料如下。

	工序	說明
工作 1	$o_{11} = (2, 4)$ $o_{12} = (1, 1)$	此工作有兩道工序，第一道需要在機器 2 執行 4 小時，第二道需要在機器 1 執行 1 小時。
工作 2	$o_{21} = (0, 2)$ $o_{22} = (2, 2)$ $o_{23} = (0, 1)$	有三道工序，第一道需要在機器 0 執行 2 小時，餘類推。
工作 3	$o_{31} = (0, 7)$	有一道工序需要在機器 0 執行 7 小時。

排程的過程說明如下：

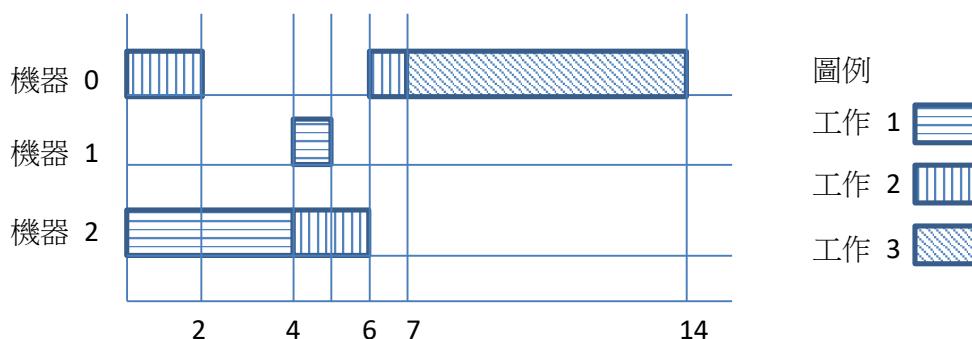
- 在開始時，每個工作都是考慮第一道工序，三個工作第 1 道工序需要的時間分別是 $t_{11} = 4$ 、 $t_{21} = 2$ 、 $t_{31} = 7$ ，這也是它們的最早完成時間，也就是 $c_{11}^* = 4$ 、 $c_{21}^* = 2$ 、 $c_{31}^* = 7$ ，因此會先排 o_{21} 。



2. 接下來，三個工作要考慮的工序分別是第 1、2、1 個工序，即 o_{11} 、 o_{22} 和 o_{31} 。
- (1) o_{11} 需要機器 2 執行 4 小時，而機器 2 可以開始加工的時間點是 0； o_{11} 沒有前一道工序。因此，這工序可以開始的時間是 $\max(0, 0) = 0$ 。是故，其最早完成時間 $c_{11}^* = \max(0, 0) + 4 = 4$ 。
 - (2) o_{22} 需要機器 2 執行 2 小時，而機器 2 可以開始加工的時間點是 0； o_{22} 前一道工序 o_{21} 的完成時間是 2。因此，這工序可以開始的時間是 $\max(0, 2) = 2$ 。是故，其最早完成時間 $c_{22}^* = \max(0, 2) + 2 = 4$ 。
 - (3) o_{31} 需要機器 0 執行 7 小時，而機器 0 可以開始加工的時間點是 2； o_{31} 沒有前一道工序。因此，這工序可以開始的時間是 $\max(2, 0) = 2$ 。是故，其最早完成時間 $c_{31}^* = \max(2, 0) + 7 = 9$ 。
- 因此，由於 c_{11}^* 和 c_{22}^* 都是最小，根據規則，工作編號小的先排，所以會排 o_{11} 。



3. 三個工作目前要考慮的工序分別第 2、2、1 個工序。依照類似的推論，我們可以得到 $c_{12}^* = 5$ ， $c_{22}^* = 6$ ， $c_{31}^* = 9$ ，因此排 o_{12} 。工作 1 的工序均已排完，所以它的完成時間是 5。
4. 剩下工作 2 與 3。 $c_{22}^* = 6$ ， $c_{31}^* = 9$ ，因此先排 o_{22} 。
5. $c_{23}^* = 7$ 而 $c_{31}^* = 9$ ，因此排 o_{23} ，工作 2 的工序已排完，所以它的完成時間是 7。
6. 剩下工作 3，因為機器 0 的下一個可以開始時間是 7， o_{31} 的完成時間是 $7+7=14$ 。



三個工作的完成時間分別是 5、7、14，所以最後輸出答案 $5+7+14=26$ 。

輸入格式

第一行有兩個整數 N 與 M ，代表 N 台機器與 M 個工作，接下來有 M 個工作的資訊，輸入的順序即是工作編號順序。每個工作資訊包含兩行，第一行是整數 P ，代表到工序數量；第二行是 $2 \cdot P$ 個整數，每兩個一組依序代表一道工序的機器編號與需求時間。機器的編號由 0 開始。參數 N 、 M 、 P 以及每個工序的需求時間都是不超過 100 的正整數。

輸出格式

輸出每個工作的完成時間的總和。

範例一：輸入

```
3 3
2
2 4 1 1
3
0 2 2 2 0 1
1
0 7
```

範例二：輸入

```
2 3
1
0 4
1
1 5
1
1 3
```

範例一：正確輸出

26

範例二：正確輸出

15

評分說明

輸入包含若干筆測試資料，每一筆測試資料的執行時間限制均為 1 秒，依正確通過測資筆數給分。其中：

第 1 子題組 20 分：只有一台機器，各工作只有一道工序。

第 2 子題組 30 分：各工作只有一道工序。

第 3 子題組 50 分：無其他限制。

第 4 題 反序數量

問題敘述

考慮一個數列 $A = (a[1], a[2], a[3], \dots, a[n])$ 。如果 A 中兩個數字 $a[i]$ 和 $a[j]$ 滿足 $i < j$ 且 $a[i] > a[j]$ ，則我們說 $(a[i], a[j])$ 是 A 中的一個反序 (inversion)。定義 $W(A)$ 為數列 A 中反序的數量。例如，在數列 $A = (3, 1, 9, 8, 9, 2)$ 中，一共有 $(3, 1)$ 、 $(3, 2)$ 、 $(9, 8)$ 、 $(9, 2)$ 、 $(8, 2)$ 、 $(9, 2)$ 一共 6 個反序，所以 $W(A) = 6$ 。

給定一個數列 A ，計算 $W(A)$ 最簡單的方法是對所有 $1 \leq i < j \leq n$ 檢查數對 $(a[i], a[j])$ ，但是在序列太長時，計算時間就會超過給定的時限。以下是運用分而治之 (divide and conquer) 的策略所設計的一個更有效率的計算方法。

1. 將 A 等分為前後兩個數列 X 與 Y ，其中 X 的長度是 $n/2$ 。
2. 遞迴計算 $W(X)$ 和 $W(Y)$ 。
3. 計算 $W(A) = W(X) + W(Y) + S(X, Y)$ ，其中 $S(X, Y)$ 是由 X 中的數字與 Y 中的數字所構成的反序數量。

以 $A = (3, 1, 9, 8, 9, 2)$ 為例， $W(A)$ 計算如下。

1. 將 A 分為兩個數列 $X = (3, 1, 9)$ 與 $Y = (8, 9, 2)$ 。
2. 遞迴計算得到 $W(X) = 1$ 和 $W(Y) = 2$ 。
3. 計算 $S(X, Y) = 3$ 。因為有三個反序 $(3, 2)$ 、 $(9, 8)$ 、 $(9, 2)$ 是由 X 中的數字與 Y 中的數字所構成。所以得到 $W(A) = W(X) + W(Y) + S(X, Y) = 1+2+3 = 6$ 。

請撰寫一個程式，計算一個數列 A 的反序數量 $W(A)$ 。

輸入格式

測試資料有兩列，第一列為一個正整數 n ，代表 A 的長度。第二列有 n 個不大於 10^6 的非負整數，代表 $a[1], a[2], a[3], \dots, a[n]$ ，數字間以空白隔開。

輸出格式

輸出 A 的反序數量 $W(A)$ 。請注意 $W(A)$ 可能會超過一個 32-bit 整數所能表示的範圍。

範例一：輸入

```
6
3 1 9 8 9 2
```

範例一：正確輸出

```
6
```

範例二：輸入

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

範例二：正確輸出

```
9
```

評分說明

輸入包含若干筆測試資料，每一筆測試資料的執行時間限制均為 1 秒，依正確通過測資筆數給分。其中：

第 1 子題組 10 分： $1 \leq n \leq 10^3$ 。

第 2 子題組 30 分： $1 \leq n \leq 10^5$ ， n 為偶數，輸入數列保證 $a[1] \leq a[2] \leq a[3] \leq \dots \leq a[m]$ 且 $a[m+1] \leq a[m+2] \leq a[m+3] \leq \dots \leq a[n]$ ，其中 $m = n/2$ 。也就是數列前半與後半是各自排好序的。

第 3 子題組 60 分： $1 \leq n \leq 10^5$ ，無其他限制。

第 1 題 運動明星

問題描述

大明是一位體壇明星，他第一次參加世界盃就排入世界前十名，之後就未曾掉出前十名榜外。除了球技高超，大明還有一個「獎金王」的稱號。每年大明都會參加各式各樣的比賽（無論比賽是否計入世界排名），並且獲得高額的比賽獎金。

熱心的網友整理了大明出道以來歷年世界前十強的獎金資料，現在請你撰寫一個程式計算大明所獲得的總獎金以及有多少年大明是前十強中獲得獎金最多的人（也就是該年度沒有其他選手獲得的獎金比大明多），證明他是名符其實的獎金王！

輸入格式

第一行有一個整數值 T ($T \leq 20$)，代表大明出賽的年度數。自第二行起有 T 行，每行代表一個年度的數據。每行有十一個整數，第一個數字 r ($1 \leq r \leq 10$) 是大明在該年度的名次，接下去十個數字是該年度前十名選手的獎金金額（以萬元為單位，數值不超過 10^5 ），依序從第一名開始到第十名，數字間以空白隔開。

輸出格式

輸出兩行，第一行是大明所獲得的總獎金數；第二行是有多少年大明獲得最高額獎金。

範例一：輸入

```
2
3 5000 900 5000 700 600 500 400 300 200 100
6 1000 900 800 700 600 999 400 300 200 100
```

範例一：正確輸出

```
5999
1
```

範例一說明：第一年大明排名第三，獎金 5000 萬元，與第一名一樣都是十位選手中最高的。第二年大明排名第六，獎金 999 萬元不是最高，最高的是第一名的 1000 萬元。所以總獎金是 5999，有 1 年得到最高額獎金。

範例二：輸入

```
5
3 1000 900 5000 700 600 500 400 300 200 100
6 1000 900 800 700 600 999 400 300 200 100
2 800 1000 900 200 135 246 778 921 876 199
4 222 222 222 222 222 222 222 222 222 222
7 119 118 117 116 115 114 113 112 111 110
```

範例二：正確輸出

```
7334
3
```

範例二說明：總獎金數是 $5000+999+1000+222+113=7334$ 。在大明生涯的第一年、第三年和第四年共有 3 年是該年度前十名中獎金金額最高的。

評分說明

輸入包含若干筆測試資料，每一筆測試資料的執行時間限制均為 1 秒，依正確通過測資筆數給分。其中：

第 1 子題組 50 分： $T = 1$ 。

第 2 子題組 50 分：無額外限制。

第 2 題 單調字串

問題描述

考慮由 A 到 Z 等二十六個大寫字母所構成的字串，如果一個字串所用到的相異大寫字母比較少，則我們稱這種字串較為單調，例如：字串 “ABBA” 比字串 “GOODY” 單調；因為前者只有兩個相異的字母，而後者有四個相異的字母。給定一正整數 N 以及 N 個字串，請寫一支程式找出最單調的字串。如果這樣的字串不只一個，則輸出依英文字典順序排序在最前的那一個最單調字串。

輸入格式

每一測試資料的第一行有一正整數 N ，代表字串的筆數。接下來 N 行，每一行有一個字串，字串的長度不超過 50。

輸出格式

輸出最單調的字串，如果這樣的字串不唯一，則依英文字典順序，輸出排在最前面的最單調字串。

範例一：輸入 4 CRAZY FOOD BABAAA GOODBYE	範例一：正確輸出 BABAAA
範例二：輸入 5 PAPA ABBB ABBA ABBA AACDW	範例二：正確輸出 ABBA

評分說明

輸入包含若干筆測試資料，每一筆測試資料的執行時間限制均為 1 秒，依正確通過測資筆數給分。其中：

第 1 子題組 40 分： $1 < N \leq 10$ ，每筆測試資料中最單調的字串只有一個，也就是不需要考慮字典順序。

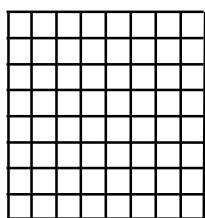
第 2 子題組 60 分： $N \leq 100$ ，無額外限制。

第 3 題 壓縮影像

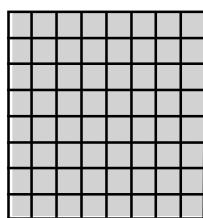
題目敘述

DF-expression 是一種壓縮黑白影像的方法。一張正方形的黑白影像可以看成是一個 $n \times n$ 的矩陣，矩陣的每一個元素是 0 或 1，用來表示一個像素是白或黑。本題中，我們假設 n 是 2 的冪次，也就是存在某個非負整數 k 使得 $n = 2^k$ 。

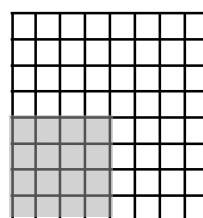
一個正方形區塊的 DF-expression 的遞迴定義如下：如果每一格像素都是白色，我們用 0 來表示，如圖(a)；如果每一格像素都是黑色，我們用 1 來表示，如圖(b)；如果並非每一格像素都同色，我們先將影像均等劃分為左上、右上、左下、右下四個小正方形(也就是四個方陣)後，然後表示如下：先寫下 2，之後依續接上左上、右上、左下、右下四塊的 DF-expression，如圖(c)和(d)。



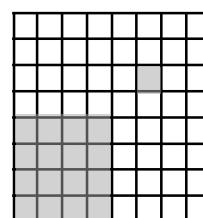
(a) 0



(b) 1



(c) 20010



(d) 2020020100010

一個壓縮後的影像會表示成一個由 0、1、2 組成的字串 S 。在這個問題中，根據字串 S 以及影像尺寸 n ，請計算原始影像中有多少個像素是 1。

輸入格式

測試資料有兩行，第一行是影像的 DF-expression S ，是由連續的 0、1、2 組成的字串，字串長度小於 $1,100,000$ 。第二行為正整數 n ， $1 \leq n \leq 1024$ ，代表影像的大小為 $n \times n$ ，其中 n 必為 2 的冪次。

輸出格式

請輸出該影像中有多少像素是 1。

範例一：輸入 2200101020110 4	範例一：正確輸出 7
範例二：輸入 2020020100010 8	範例二：正確輸出 17

評分說明

對每一筆測試資料的執行時間限制均為 1 秒，，依正確通過測資筆數給分。其中：

第 1 子題組 10 分： $n = 2$ 。

第 2 子題組 20 分： $n = 4$ 。

第 3 子題組 70 分： $1 \leq n \leq 1024$ 。

第 4 題 置物櫃出租

問題敘述

王老先生有一個置物櫃，共有 M 個相同大小的格子，置物櫃目前已經租給 n 個客戶，第 i 位客戶所租的大小為 $f(i)$ 個格子 ($1 \leq i \leq n$)。目前的承租量總和不超過 M ，但是現在王老先生自己需要使用 S 個格子的置物櫃，如果剩下的容量不夠，就必須退掉某些客戶的租約。假設每個客戶所租容量只能全退或全不退，而退租第 i 個客戶損失的利益與其所租容量 $f(i)$ 相同，請寫一支程式計算王老先生最小的損失利益，如果不須退租，則損失為 0。

輸入格式

測試資料有兩行，第一行有三個正整數，依序是 n 、 M 與 S ，其中 $S \leq M$ ，第二行是 n 個正整數 $f(1), f(2), f(3), \dots, f(n)$ ，同一行的數字間以空白隔開。

輸出格式

輸出王老先生最小的損失利益。

範例一：輸入	範例一：正確輸出
3 10 6 4 4 1	5
範例二：輸入	範例二：正確輸出
5 20 14 8 2 7 2 1	15

評分說明

每一筆測試資料的執行時間限制均為 2 秒，依正確通過測資筆數給分。其中：

第 1 子題組 20 分： $1 \leq n \leq 10$ ， $M \leq 10000$ 。

第 2 子題組 30 分： $1 \leq n \leq 100$ ， $M \leq 50000$ 。

第 3 子題組 50 分： $1 \leq n \leq 100$ ， $M \leq 200000$ 。

第 1 題 車站遊戲

問題描述

小花正在玩一個捷運車站的遊戲，請你撰寫一個程式計算小花在遊戲中的總得分。遊戲中有 1 號、2 號與 3 號共三個車站。一開始小花會被指定到其中一個車站，每一回合，小花到達指定的車站後可以抽一個幸運數字，根據所在的車站以及抽到的幸運數字 L 可以得到一個分數，並且指示下一回合要去的車站。規則如下：

- 目前在 1 號車站：可以得到 L 分。如果 L 是偶數，下一回合須前往 2 號車站；否則，繼續留在 1 號車站。
- 2 號車站：可以得到 $2L$ 分。如果 L 是 3 的倍數，下一回合須前往 3 號車站；否則，前往 1 號車站。
- 3 號車站：得分等於 L 除以 10 的商。如果 L 是 5 的倍數，下一回合繼續留在 3 號車站；否則，前往 1 號車站。

輸入格式

第一行有兩個正整數 n 與 s ， $n \leq 20$ 是遊戲的回合數而 s 是開始的車站編號。第二行有 n 個數字，代表小花每一回合依序抽到的幸運數字，每個幸運數字都是不超過 100 的非負整數，同一行的數字間都以空白隔格。

輸出格式

輸出小花的總得分。

範例一：輸入 1 2 3	範例一：正確輸出 6
範例一說明：玩 1 回合，從 2 號車站開始，抽到 3，得到 6 分。	
範例二：輸入 4 1 1 2 3 12	範例二：正確輸出 10
範例二說明：玩 4 回合，第一回合從 1 號車站開始，抽到 1，得到 1 分，然後留在 1 號車站；第二回合抽到 2，得 2 分，然後前往 2 號車站；第三回合抽到 3，得 6 分，然後前往 3 號車站；第四回合抽到 12，得 1 分。遊戲結束，一共得到 10 分。	

評分說明

每一筆測試資料的執行時間限制均為 1 秒，依正確通過測資筆數給分。其中：

第 1 子題組 30 分：已知 $n = 1$ 。

第 2 子題組 70 分：無額外限制。

第 2 題 紅白彩帶

問題描述

小軒的美術老師給了他紅色蠟筆與一條彩帶，彩帶上被分成 n 個相同大小的格子，有的格子是紅色，有些則是白色。小軒拿到彩帶後開始塗顏色，每次會將一個白色格子塗滿紅色，然後小軒會算一算目前最長與最短紅色區域的長度(也就是佔了幾格)，兩個相鄰的紅色格子就屬於同一個紅色區域。小軒一共塗了 k 次，請你計算出每一次紅色區域的最大與最小長度，並輸出個別的總和。

小軒的彩帶可以看成一個一維陣列，以 0 表示白色，而 1 表示紅色。彩帶的格子從 1 開始由左而右依序編號，小軒每次將某個格子塗上紅色。以下是一個例子。

格子編號	1	2	3	4	5	6	7	8	9
剛開始的彩帶	0	1	1	0	0	1	0	1	0

一開始紅色區域最大的長度是 2，最小的長度是 1。

第1次塗在第5格後	0	1	1	0	1	1	0	1	0
-----------	---	---	---	---	---	---	---	---	---

第1次塗完後最長的紅色區域有 2 塊，長度都是 2，最小的長度是 1。

第2次塗在第1格後	1	1	1	0	1	1	0	1	0
-----------	---	---	---	---	---	---	---	---	---

第2次塗完後紅色區域的最大長度是 3，最小長度是 1。

第3次塗在第7格後	1	1	1	0	1	1	1	1	0
-----------	---	---	---	---	---	---	---	---	---

第3次塗完後紅色區域的最大長度是 4，最小長度是 3。

以這個例子而言，每一次(包含剛開始) 紅色區域的最大長度總和是 $2 + 2 + 3 + 4 = 11$ ；而紅色區域的最小長度總和是 $1 + 1 + 1 + 3 = 6$ 。

輸入格式

輸入有三行，第一行是兩個整數，依序是 n 與 k ，代表彩帶長度以及小軒塗色的次數， $n \leq 100,000$ 、 $k \leq 20,000$ 。第二行有 n 個 0 或 1 的數字，依序代表彩帶第 1 格至第 n 格的顏色，0 代表白色，1 代表紅色。第三行有 k 個數字，依序代表每一次塗紅色的格子編號，若 $k=0$ ，則第三行為空行。同一行數字之間都是以空白間隔。小軒每次一定都塗在白色格子上，而且最大的紅色區域長度不會超過 10,000。

輸出格式

輸出兩行，第一行是每次紅色區域最大長度的總和，第二行是每次紅色區域最小長度的總和。

範例一：輸入	範例一：正確輸出
5 1 1 0 1 0 1 2	4 2
範例一說明：一開始最大長度是 1 而最小長度也是 1，塗在第二格之後最大長度是 3	

而最小長度還是 1。最大長度總和為 $1+3=4$ ；最小長度總和為 $1+1=2$ 。

範例二：輸入

9 3
0 1 1 0 0 1 0 1 0
5 1 7

範例二：正確輸出

11
6

範例二說明：此為題目中所舉的例子。

評分說明

每筆測試資料執行時間限制為 1 秒，依正確通過測資筆數給分，其中：

第 1 子題組 20 分： $n \leq 1,000$ 、 $k = 0$ 。

第 2 子題組 50 分： $n \leq 10,000$ 、 $k \leq 1,000$ 。

第 3 子題組 30 分：沒有額外限制。

第 3 題 合成函數

問題描述

棟棟正在學習多變量函數的合成，他的數學課本上列出三種函數 $f(x)$ 、 $g(x, y)$ 與 $h(x, y, z)$ ，其中 $f(x) = 2x - 3$ ； $g(x, y) = 2x + y - 7$ ； $h(x, y, z) = 3x - 2y + z$ 。比如說， $f(5) = 2 \times 5 - 3 = 7$ ， $g(3, 4) = 2 \times 3 + 4 - 7 = 3$ ，因此 $h(f(5), g(3, 4), 3) = h(7, 3, 3) = 3 \times 7 - 2 \times 3 + 3 = 18$ 。棟棟學會了這個概念，想把算式傳給同學，他突發奇想，將所有的左括號與逗點都用空白取代而右括弧直接刪除，他認為這樣還是可以看得懂的。例如上述 $h(f(5), g(3, 4), 3)$ 這個式子被他簡化成

$hf5\ g\ 3\ 4\ 3$

其中函數與數字間，皆有一個空白隔開。事實上，棟棟的想法沒有錯，棟棟的資訊老師告訴他，這樣將所有括弧與逗點以空白取代的方式還是可以唯一確定原來的計算式，而且可以用遞迴的方式求出函數的值。本題請你撰寫一支程式計算這樣的一個合成函數之值。

輸入格式

輸入只有一行，亦即一條簡化式，輸入行的總長度包含空白不超過 1000 個字元。簡化式中只有 f 、 g 、 h 三種字元與一些整數，字母表示所對應之函數，字元或整數間皆以一個空白隔開，輸入的第一個字元必然不是數字，而輸入之整數其絕對值不超過 1000。

輸出格式

輸出簡化式之合成函數的值，你可以假設答案與運算過程的數值皆在 $-10^9 \sim 10^9$ 的範圍內。

範例一：輸入	範例一：正確輸出
$f\ f\ f\ 2$	-5
範例一說明：原始算式以及計算方式為 $f(f(f(2))) = f(f(1)) = f(-1) = -5$ 。	
範例二：輸入	範例二：正確輸出
$h\ f\ 5\ g\ 3\ 4\ 3$	18
範例二說明：此輸入即為題目敘述中的範例。	

評分說明

每筆測試資料執行時間限制為 1 秒，依正確通過測資筆數給分，其中：

第 1 子題組 10 分：簡化式中函數只有 f 而整數皆為非負的一位數。

第 2 子題組 40 分：簡化式中的數字都是非負的一位數。

第 3 子題組 50 分：沒有額外限制。

第 4 題 可挑戰人數

問題描述

有 n 個人從 1 到 n 排名，以下以排名當作每個人的編號來稱呼他們。每個人有兩項被考慮的特質： i 的基本能力值為 $b(i)$ 而潛力值為 $p(i)$ 。定義 i 的「無法超越的位置」為：從 i 往前找，所碰到第一個 $b(j) > b(i) + p(i)$ 的位置 j ，而 $i - j - 1$ 則稱為 i 的「最大可挑戰人數」 $S(i)$ ；如果 i 的前面所有人的基本能力值都不大於 $b(i) + p(i)$ ，則最大可挑戰人數 $S(i) = i - 1$ ，也就是排在他之前全部的人數。

舉例來說，假設 $n = 5$ ，基本能力值 b 依序為 $(5, 4, 1, 1, 3)$ 而潛力值 p 依序是 $(0, 0, 4, 0, 1)$ 。在這個例子中，對排名第 1 的人而言，前面沒有人，所以 1 的最大可挑戰人數 $S(1) = 0$ 。對排名第 2 的人而言， $b(2) + p(2) = 4 + 0 = 4$ ，而往前看到第一個大於 4 的是 $b(1)=5$ ，所以 $S(2) = 2 - 1 - 1 = 0$ 。對排名第 3 的人而言， $b(3) + p(3) = 1 + 4 = 5$ ，前面沒有人的基本能力值 b 大於 5，所以 $S(3) = 3 - 1 = 2$ 。而 $b(4) + p(4) = 1 + 0 = 1$ ， $b(3) = 1$ 不大於 1， $b(2)=4 > 1$ ，所以位置 2 是排名第 4 的無法超越位置， $S(4) = 4 - 2 - 1 = 1$ 。同理可求出 $S(5) = 3$ ，因為他的不可超越位置是排名第 1 的基本能力值 5。

輸入所有人的基本能力值與潛力值，請計算所有人最大可挑戰人數 $S(i)$ 的總和。以上面的例子而言，所求答案是 $0 + 0 + 2 + 1 + 3 = 6$ 。

輸入格式

每筆測試資料有三行，第一行為人數 n ， $n \leq 2 \times 10^5$ ；第二行有 n 個正整數，依序代表每個人的基本能力值 $b(1), b(2), \dots, b(n)$ ；第三行有 n 個非負整數，依序代表每個人的潛力值 $p(1), p(2), \dots, p(n)$ ；每一個基本能力值與潛力值都不超過 10^7 ，同一行數字之間都是以空白間隔。

輸出格式

輸出所有最大可挑戰人數 $S(i)$ 的總和。注意，答案可能超過 2^{32} ，但不會超過 2^{60} 。

範例一：輸入 8 1 2 2 3 4 4 3 1 0 0 0 0 0 0 0 0	範例一：正確輸出 15
範例二：輸入 5 5 4 1 1 3 0 0 4 0 1	範例二：正確輸出 6

評分說明

對於每一筆測試資料，Python 程式的執行時間限制為 2 秒，其餘程式語言的執行時間限制為 0.5 秒，依正確通過測資筆數給分。其中：

第 1 子題組 20 分： $1 \leq n \leq 10^3$ 且所有潛力值 $p(i)$ 皆為 0。

第 2 子題組 40 分：所有潛力值 $p(i)$ 皆為 0。

第 3 子題組 40 分：無額外限制。

第 1 題 籃賽短訊

問題描述

網路社群風行，多數的運動賽事聯盟都會經營自己的社群，即時張貼比賽資訊，以維持粉絲對賽事的關注度。APCS 籃球聯盟總決賽正在如火如荼的進行中，聯盟希望社群小編能在比賽一結束就立刻以短訊發布比賽結果。社群小編求助於你，希望你能撰寫一個程式，讀入今天兩場籃賽的四節分數，自動產生短訊內容。詳細輸出內容與格式，請參見以下說明。

輸入格式

輸入有四行，前兩行依序代表第一場比賽中主隊與客隊獲得的分數，後兩行依序代表第二場比賽中主隊與客隊獲得的分數。這四行每行固定有四個小於 100 的非負整數，以空格隔開，代表比賽中四節的分數，將四節分數相加就是該球隊在該場比賽的總分。假設每場比賽必定會分出勝負，不會有兩隊同分的情況。

輸出格式

輸出三行，第一行是主隊與客隊在第一場比賽的總分情形(格式請見範例說明)，第二行是主隊與客隊在第二場比賽的總分情形，第三行是兩場比賽後的勝負情形，如果主隊兩戰皆勝則輸出 Win；如果主隊兩戰皆敗則輸出 Lose；否則輸出 Tie，請注意大小寫。

範例一：輸入	範例一：正確輸出
20 15 25 20 19 11 24 16 21 12 23 15 19 20 20 10	80:70 71:69 Win

範例一說明

第一場比賽主隊的總得分是 $20 + 15 + 25 + 20 = 80$ ，客隊的總得分是 $19 + 11 + 24 + 16 = 70$ ，所以第一行輸出的是第一場比數為 80:70。類似地，第二場比賽主隊得分為 $21 + 12 + 23 + 15 = 71$ ，客隊得分為 $19 + 20 + 20 + 10 = 69$ ，第二場比數為 71:69。最後，因為兩場皆為主隊獲勝，所以第三行輸出 Win。

範例二：輸入	範例二：正確輸出
10 10 20 20 15 14 16 22 20 22 23 15 20 20 14 16	60:67 80:70 Tie

範例三：輸入	範例三：正確輸出
15 15 20 20	70:72
22 14 16 20	78:80
19 21 23 15	Lose
20 10 34 16	

評分說明

輸入包含若干筆測試資料，每一筆測試資料的執行時間限制均為 1 秒，依正確通過測資筆數給分，滿分 100 分。

第 2 題 機器人走棋盤

問題描述

有一個方格棋盤，棋盤的每一個格子裡都標示了一個整數，而且這些整數都是不相同的。現在有一個機器人在此方格棋盤上行動，每一次機器人只會移動到目前所在位置的上下左右四個相鄰格子中的一格。機器人的起點是數字最小的格子，每次移動會在可以走的位置中挑選數字最小的格子，但是機器人永遠都不會走到他曾經走過的格子，當然他也不會走到範圍之外。當無路可走的時候，機器人就會停下來。輸入方格棋盤中每個格子的數字，請模擬機器人走過的路徑，並輸出機器人走過的格子的數字總和。

以下是一個例子，輸入的 4×5 的方格內的數字如圖中所標示。在本例子中，機器人的起點會是 1，所走的路徑是 $1 \rightarrow 4 \rightarrow 6 \rightarrow 7 \rightarrow 13 \rightarrow 20 \rightarrow 21 \rightarrow 29 \rightarrow 30$ 。走到 30 的時候已經無路可走，所以機器人就停止了，而經過的數字總和是 131。

25	7	13	20	30
11	6	4	21	29
2	8	1	35	40
3	9	5	12	15

輸入格式

輸入的第一行是兩個不超過 100 的正整數 m 與 n ，代表是一個 $m \times n$ 的方格棋盤，接下來有 m 行，每行 n 個數字，分別是方格棋盤由上而下，由左而右的數字。方格內的數字皆為不超過 100,000 的非負整數，同一行數字之間以空白間隔。

輸出格式

輸出機器人走過的格子中數字的總和。

範例一：輸入 1 7 6 8 7 2 1 4 5	範例一：正確輸出 24
範例二：輸入 4 5 25 7 13 20 30 11 6 4 21 29 2 8 1 35 40 3 9 5 12 15	範例二：正確輸出 131

評分說明

輸入包含若干筆測試資料，每一筆測試資料的執行時間限制均為 1 秒，依正確通過測資筆數給分。其中：

第 1 子題組 30 分： $m = 1$ ，也就是只有一列， $1 \leq n \leq 100$ 。

第 2 子題組 70 分： $1 \leq n, m \leq 100$ 。

第 3 題 卡通團隊

問題描述

小美有 m 個喜歡的卡通人物， $2 \leq m \leq 26$ ，這些卡通人物分別以前 m 個英文大寫字母代表。她正在玩一個組團的遊戲，每次把其中若干個人物的代碼寫在一起成為一個字串就算組成一個團隊，一個字母如果出現在一個字串中，就代表該卡通人物出現在此團隊內，兩個團隊的成員如果完全相同就算相同的團隊，不管字母的順序以及是否有字母重複出現，例如「AKB」、「KAKB」、「BKAAB」與「BAK」都是代表相同的團隊，但「AKB」與「AKBF」並非相同團隊。

如果兩個團隊沒有相同的成員而且聯集起來是所有 m 個卡通人物，則這兩個團隊稱為「互補團隊」。輸入 m 以及小美所組的 n 個團隊，請計算有幾對是互補團隊。我們假設小美並不會組兩個相同的團隊。

輸入格式

輸入的第一行是兩個整數 m 與 n ， $2 \leq m \leq 26$ ， $1 \leq n \leq 50000$ 。第二行開始有 n 行，每行一個字串代表一個團隊，每個字串的長度是不超過 100 的正整數，每個字串必定只由前 m 個大寫英文字母組成。

輸出格式

輸出小美有多少對互補團隊。

範例一：輸入	範例一：正確輸出
10 5 AJBA HCEFGGC BIJDAIJ EFCDHGI HCEFGA	2

範例一說明：

有十個人物，所以代表字母為 ABCDEFGHIJ，在五個團隊中，AJBA 與 EFCDHGI 是互補團隊，因為沒有相同成員且聯集起來是全部 10 個人物，請留意字串中可能有重複出現的字母，例如 AJBA 其實是與 ABJ 相同的團隊。此外，HCEFGGC 與 BIJDAIJ 也是互補團隊，總共有兩對互補團隊。

範例二：輸入	範例二：正確輸出
11 5 AJBBJ HCEFGK BIJDA EFCDHGICDH EFCJHGICDKC	1

評分說明

對於每一筆測試資料，Python 程式的執行時間限制為 2 秒，其餘程式語言的執行時間限制為 1 秒，依正確通過測資筆數給分。其中：

第 1 子題組 20 分： $n \leq 500$ 。

第 2 子題組 80 分： 沒有其他限制。

第 4 題 完美彩帶

問題描述

有一條細長的彩帶，彩帶區分成 n 格，每一格的長度都是 1，每一格都有一個顏色，相鄰格子的顏色可能相同或不同，已知總共有 m 種不同的顏色。在這個彩帶中，如果有某一段長度為 m 的區段剛好 m 種顏色都各出現一次，則我們稱這一個區段的彩帶是一段「完美彩帶」。輸入 m 以及彩帶每一格的顏色編號，請找出總共有多少段可能的完美彩帶。

舉例來說，以下是一長度 $n = 10$ 而顏色數 $m = 4$ 的彩帶，其中每一個數字代表一種顏色的編號，請注意，顏色的編號並不一定是由 0 開始的連續整數：

格子編號	1	2	3	4	5	6	7	8	9	10
顏色編號	1	4	1	7	6	4	4	6	1	7

在本例中，區間 $[2, 5]$ 是一段完美彩帶，因為顏色 4、1、7、6 剛好各出現一次，此外，區間 $[3, 6]$ 與 $[7, 10]$ 也都是完美彩帶，所以總共有三段可能的完美彩帶。請注意，兩段完美彩帶之間可能重疊。

輸入格式

每筆測試資料有兩行，第一行為整數 m 和 n ，滿足 $2 \leq m \leq n \leq 2 \times 10^5$ ；第二行有 n 個以空白間隔的數字，依序代表彩帶從左到右每一格的顏色編號，顏色編號是不超過 10^9 的非負整數，每一筆測試資料的顏色數量必定恰好為 m 。

輸出格式

輸出總共有多少段完美彩帶。

範例一：輸入	範例一：正確輸出
3 7 1 2 2 1 0 1 2	2
範例二：輸入	範例二：正確輸出
4 10 1 4 1 7 6 4 4 6 1 7	3

評分說明

輸入包含若干筆測試資料，每一筆測試資料的執行時間限制均為 1 秒，依正確通過測資筆數給分。其中：

第 1 子題組 20 分： $m \leq 100 \leq n \leq 10^3$ 且顏色編號恰為 $0 \sim m - 1$ 。

第 2 子題組 50 分： $2 \leq m \leq n \leq 2 \times 10^5$ 且顏色編號恰為 $0 \sim m - 1$ 。

第 3 子題組 30 分：沒有其他限制。

大學程式設計先修檢測

2019.10.26

實作題 注意事項

1. 請依題目敘述撰寫一完整程式，上傳送審之程式檔案副檔名必須為 .c, .cpp, .java, 或.py，如下表所示。另若以 Java 撰寫程式，class 名稱必須與檔名 (P1, P2, P3, 或P4) 一致。

	C	C++	Java	Python
樂曲修改	P1.c	P1.cpp	P1.java	P1.py
洗牌	P2.c	P2.cpp	P2.java	P2.py
闖關路線	P3.c	P3.cpp	P3.java	P3.py
刪除邊界	P4.c	P4.cpp	P4.java	P4.py

2. 上傳程式檔案前，請自行測試程式是否能正常編譯，程式執行時輸入、輸出格式是否正確。評分時，若程式無法正常編譯或執行，將以0分計算。
3. 程式內不應有開檔、讀檔、寫檔等程序，資料讀取皆應來自標準輸出入，資料輸出入格式詳各題說明。評分時，系統將自動導入測試資料，每行輸入資料最後一定有換行 (UNIX 格式)。
4. 應測生可於作答系統檢視其程式以題本上範例測試資料測試之結果，惟該結果並非該題獲得之分數。檢測結束後，將另以正式測試資料評定分數。

第 1 題 樂曲修改

問題描述

阿明是一個程式設計新手，他想嘗試運用程式來產生一首樂曲。礙於他的能力有限，他的程式只會產生一串音階而沒有節奏，每一個音階以一個正整數來代表，所以一段樂曲只是簡單的一個整數序列，例如以 1 2 3 4 5 6 7 來表示 Do Re Mi Fa Sol La Ti，而 8 代表更高音的 Do，9 代表更高音的 Re，以此類推。

朋友們聽了阿明程式創作的樂曲，發現有時候連續兩個音的音階相差太大，導致聽來不太舒服。因此阿明想要修正原本的樂曲，如果以 y_i 為第 i 個音的代表數字，他想要做的修改規則如下：

從第 4 個音 ($i=4$) 開始，由前往後依序對每一個 y_i 進行以下檢查修改：若 y_i 和 y_{i-1} 差距超過 5 (即 $|y_i - y_{i-1}| > 5$)，則將 y_i 修改成前三個音 (y_{i-3} 、 y_{i-2} 、 y_{i-1}) 的中位數，也就是將 y_i 改成「 y_{i-3} 、 y_{i-2} 、 y_{i-1} 三個數字中第二大的數字」。

舉例來說，若原曲為：

y_1	y_2	y_3	y_4	y_5	y_6
2	2	4	10	3	9

從第四個音開始檢查，因為 $|y_4 - y_3| = 10 - 4 = 6 > 5$ ，所以用前三個音 (2、2、4) 的中位數 (2) 來取代。接著檢查第五個音，原曲中第五個音和第四個音相差 7，但第四個音已經修改成 2 後，兩者只差 1，因此第五個音就不需要修改了。最後檢查第 6 個音，9 與 3 相差超過 5，所以要將第六個音改成 (4、2、3) 的中位數 (3)。整個序列修改後的結果如下：

y_1	y_2	y_3	y_4	y_5	y_6
2	2	4	2	3	3

阿明希望你幫忙撰寫一個程式來修改他的樂曲。請注意，前三個音不需要修改。

輸入格式

第一行有一個正整數 N ($4 \leq N \leq 32$)，代表這段樂曲中有幾個音。

第二行有 N 個正整數 y_i ($1 \leq i \leq N$ ， $1 \leq y_i \leq 35$)，彼此間以空白隔開。

輸出格式

輸出一行，共有 N 個正整數，代表修改後樂曲的每個音，彼此間以一個空白隔開。

範例一：輸入

4
1 10 2 8

範例一：正確輸出

1 10 2 2

範例二：輸入

8
1 7 8 8 9 16 17 26

範例二：正確輸出

1 7 8 8 9 8 8 8

範例三：輸入

9
2 1 4 10 10 1 4 7 1

範例三：正確輸出

2 1 4 2 2 1 4 7 4

評分說明

輸入包含若干筆測試資料，每一筆測試資料的執行時間限制均為 1 秒，依正確通過測資筆數給分。其中：

第 1 子題組 30 分： $N=4$ ，即樂曲只有 4 個音，你只需要考慮第四個音（如範例一）。

第 2 子題組 20 分： $y_i \leq y_{i+1}$ ，也就是輸入的音是由低到高排列（如範例二）。

第 3 子題組 50 分：無特別限制。

大學程式設計先修檢測

2019.10.26

實作題 注意事項

1. 請依題目敘述撰寫一完整程式，上傳送審之程式檔案副檔名必須為 .c, .cpp, .java, 或.py，如下表所示。另若以 Java 撰寫程式，class 名稱必須與檔名 (P1, P2, P3, 或P4) 一致。

	C	C++	Java	Python
樂曲修改	P1.c	P1.cpp	P1.java	P1.py
洗牌	P2.c	P2.cpp	P2.java	P2.py
闖關路線	P3.c	P3.cpp	P3.java	P3.py
刪除邊界	P4.c	P4.cpp	P4.java	P4.py

2. 上傳程式檔案前，請自行測試程式是否能正常編譯，程式執行時輸入、輸出格式是否正確。評分時，若程式無法正常編譯或執行，將以0分計算。
3. 程式內不應有開檔、讀檔、寫檔等程序，資料讀取皆應來自標準輸出入，資料輸出入格式詳各題說明。評分時，系統將自動導入測試資料，每行輸入資料最後一定有換行 (UNIX 格式)。
4. 應測生可於作答系統檢視其程式以題本上範例測試資料測試之結果，惟該結果並非該題獲得之分數。檢測結束後，將另以正式測試資料評定分數。

第 2 題 洗牌

問題描述

我們用整數 $1, 2, \dots, n$ 來代表 n 張不同的牌，其中 $n \geq 4$ 是一個偶數。令 $P = (p_1, p_2, \dots, p_n)$ 為一個由這 n 張牌所構成的排列。我們定義三種操作如下：

洗牌： 將 P 轉換成 $(p_1, p_{n/2+1}, p_2, p_{n/2+2}, \dots, p_{n/2}, p_n)$ ；也就是說將牌平分為兩疊後，再交錯排列；

反洗牌： 將 P 轉換成 $(p_1, p_3, p_5, \dots, p_{n-1}, p_2, p_4, p_6, \dots, p_n)$ ；

k -切牌： 將 P 轉換成 $(p_{k+1}, p_{k+2}, \dots, p_n, p_1, p_2, \dots, p_k)$ ，其中 $3 \leq k < n$ 。

例如： $P = (1, 2, 3, 4, 5, 6)$ 經洗牌後會變成 $(1, 4, 2, 5, 3, 6)$ ；再經4-切牌後會變成 $(3, 6, 1, 4, 2, 5)$ ，再經反洗牌後會變成 $(3, 1, 2, 6, 4, 5)$ 。

一個排列 P 經過一連串的操作後，會轉換成一個新的排列 X 。輸入 X 和一連串當初的操作，本題要找出原始的排列 P 。以上面的例子來說，若輸入操作後的排列 $X = (3, 1, 2, 6, 4, 5)$ 以及當初的操作順序（洗牌、4-切牌、反洗牌），則所求的原始排列就是 $P = (1, 2, 3, 4, 5, 6)$ 。

提示：依原始操作的相反順序，逐步將當初的操作還原。

輸入格式

每筆測試資料有三行，第一行為一個偶數 n ($4 \leq n \leq 200$) 以及一個整數 m ($1 \leq m \leq 20$)；第二行有 n 個正整數，代表排列 X 的內容 x_1, x_2, \dots, x_n 。第三行有 m 個整數，依序代表 X 所經過的 m 個操作，其中整數 1 代表洗牌、整數 2 代表反洗牌、整數 $k \geq 3$ 代表 k -切牌。同一行數字之間以空白間隔。

輸出格式

輸出原始的排列 P ，數字間以一個空白間隔。

範例一：輸入

6 3
3 1 2 6 4 5
1 4 2

範例一：正確輸出

1 2 3 4 5 6

範例二：輸入

6 2
5 1 3 4 6 2
1 2

範例二：正確輸出

5 1 3 4 6 2

評分說明

輸入包含若干筆測試資料，每一筆測試資料的執行時間限制均為 1 秒，依正確通過測資筆數給分。其中：

第 1 子題組 30 分： m 個操作都是 k -切牌。

第 2 子題組 70 分：無額外限制。

大學程式設計先修檢測

2019.10.26

實作題 注意事項

1. 請依題目敘述撰寫一完整程式，上傳送審之程式檔案副檔名必須為 .c, .cpp, .java, 或.py，如下表所示。另若以 Java 撰寫程式，class 名稱必須與檔名 (P1, P2, P3, 或P4) 一致。

	C	C++	Java	Python
樂曲修改	P1.c	P1.cpp	P1.java	P1.py
洗牌	P2.c	P2.cpp	P2.java	P2.py
闖關路線	P3.c	P3.cpp	P3.java	P3.py
刪除邊界	P4.c	P4.cpp	P4.java	P4.py

2. 上傳程式檔案前，請自行測試程式是否能正常編譯，程式執行時輸入、輸出格式是否正確。評分時，若程式無法正常編譯或執行，將以0分計算。
3. 程式內不應有開檔、讀檔、寫檔等程序，資料讀取皆應來自標準輸出入，資料輸出入格式詳各題說明。評分時，系統將自動導入測試資料，每行輸入資料最後一定有換行 (UNIX 格式)。
4. 應測生可於作答系統檢視其程式以題本上範例測試資料測試之結果，惟該結果並非該題獲得之分數。檢測結束後，將另以正式測試資料評定分數。

第 3 題 闖關路線

問題描述

某個闖關遊戲上有一隻皮卡丘與兩個可控制皮卡丘左右移動的按鍵。皮卡丘被安置在僅可左右移動的滑軌上。滑軌被等分成 n 個位置，由左到右分別以數字 $0 \sim n - 1$ 表示。當遊戲開始時，皮卡丘會回歸到開始位置 0，螢幕上會顯示 3 個數字： P 、 L 與 R ，其中 P 表示闖關皮卡丘所須移至的目標位置， L 與 R 則分別表示每按一次左鍵或右鍵後，皮卡丘會往左或往右移動的格子數。此外，每一個位置 x 都對應一個瞬間移動位置 $S(x)$ ；每一次按鍵後，皮卡丘會先依據按鍵往左或右移動到某個位置 x ，接著瞬間移動至 $S(x)$ 。某些點的瞬間移動位置等同原地點，也就是 $S(x) = x$ ，這些點稱為停留點。開始與目標位置都一定是停留點；此外，每個點的瞬間移動位置都一定是停留點（除非超出界外），也就是不會發生連續瞬間移動的情形。

遊戲的目標是以左右鍵操作皮卡丘由開始位置到達目標位置，但必須以最少的按鍵數到達目標位置才算闖關成功。此外，在移動過程中不可以超過滑軌的範圍，如果某次按鍵後皮卡丘要移動到的位置超過滑軌的範圍 $[0, n - 1]$ ，則算闖關失敗；某些點的瞬間移動位置也可能會超出滑軌的範圍，移動到這些點也會導致闖關失敗。

輸入格式

輸入有兩行，第一行有 4 個數字，第 1 個為 n ，第 2 個為目標位置 P ，第 3 個為按下左鍵後往左移動的格子數 L ，第 4 個為按下右鍵後往右移動的格子數 R ，後三個數字皆為小於 n 之正整數，且 $2 \leq n \leq 10^6$ 。第二行有 n 個整數，依序是各點的瞬間移動位置 $S(0), S(1), \dots, S(n - 1)$ ，這些數字是絕對值不超過 10^8 的整數。

輸出格式

輸出到達目標位置所需的最少按鍵數，如果無法到達目標位置，則輸出 -1 。

範例一：輸入

5 3 1 2
0 3 2 3 5

範例一：正確輸出

2

範例一說明：

此範例中，位置區間為 $[0, 4]$ ，目標位置為 3，每按一次左鍵往左移動 1 格，每按一次右鍵往右移動 2 格。到目標位置最少的按鍵數是 2 次：從起點 0 第一次按下右鍵會到達位置 2，因為 $S(2)=2$ ，因此停留在 2。第二次按鍵如果按下右鍵會到達 4，因為 $S(4)=5$ 會出界，闖關失敗；如果第二次是按下左鍵，會往左移一格到達 1，因為 $S(1)=3$ ，所以瞬間移動到 3，由於 3 是目標位置，所以就闖關成功了。過程中按了兩次按鍵，這是最少的按鍵數，所以輸出答案是 2。

範例二：輸入

```
10 8 2 3  
0 5 2 3 4 5 6 6 8 9
```

範例二：正確輸出

```
3
```

範例二說明：

此範例中，位置區間為 $[0, 9]$ ，目標位置為 8，按一次左鍵往左移動 2 格，按一次右鍵往右移動 3 格。到達目標位置的最少按鍵數是 3 次，其經過的路徑是：

0，按右鍵 \rightarrow 3（停留在 3），按左鍵 \rightarrow 1（瞬間移動到 5），按右鍵 \rightarrow 8（停留在 8，到達）。

評分說明

輸入包含若干筆測試資料，對於每一筆測試資料，Python 程式的執行時間限制為 2 秒，其餘程式語言的執行時間限制為 1 秒，依正確通過測資筆數給分。其中：

第 1 子題組 20 分： $n \leq 100$ 。

第 2 子題組 20 分： $n \leq 1000$ 。

第 3 子題組 60 分：無額外限制。

大學程式設計先修檢測

2019.10.26

實作題 注意事項

1. 請依題目敘述撰寫一完整程式，上傳送審之程式檔案副檔名必須為 .c, .cpp, .java, 或.py，如下表所示。另若以 Java 撰寫程式，class 名稱必須與檔名 (P1, P2, P3, 或P4) 一致。

	C	C++	Java	Python
樂曲修改	P1.c	P1.cpp	P1.java	P1.py
洗牌	P2.c	P2.cpp	P2.java	P2.py
闖關路線	P3.c	P3.cpp	P3.java	P3.py
刪除邊界	P4.c	P4.cpp	P4.java	P4.py

2. 上傳程式檔案前，請自行測試程式是否能正常編譯，程式執行時輸入、輸出格式是否正確。評分時，若程式無法正常編譯或執行，將以0分計算。
3. 程式內不應有開檔、讀檔、寫檔等程序，資料讀取皆應來自標準輸出入，資料輸出入格式詳各題說明。評分時，系統將自動導入測試資料，每行輸入資料最後一定有換行 (UNIX 格式)。
4. 應測生可於作答系統檢視其程式以題本上範例測試資料測試之結果，惟該結果並非該題獲得之分數。檢測結束後，將另以正式測試資料評定分數。

第 4 題 刪除邊界

問題描述

在本題中，輸入的是一個 m 列 n 行的矩陣，矩陣的內容皆為 0 或 1。一個矩陣的第一列與最後一列以及第一行與最後一行稱為該矩陣的四條邊界線，如果某一條邊界線的內容都是相同元素（全是 0 或全是 1），則可以刪除該邊界線。如果一條邊界線的內容不完全相同，你可以修改某些格子的內容讓邊界線的內容變成相同後，再刪除該邊界線。矩陣在刪除一條邊界線後，還是一個矩陣，但列數或行數會減少一，本題的目標是重複執行修改與刪除邊界的動作，最後將整個矩陣刪除。輸入一個矩陣，請計算最少要修改多少個元素才能將整個矩陣刪除。請注意：根據定義，只有一個元素的邊界線是可以被刪除的，所以只有一列或是只有一行的矩陣可以不需要任何修改就可以被全部刪除。

以下是一個例子，(1) 是一個原始輸入的 4×5 矩陣 M ，經過 (2) ~ (6) 的步驟後，只剩下一個 1×3 矩陣，而這個只有一列的矩陣不需修改就可以被刪除。

1	1	1	0	1
0	1	0	1	1
0	0	0	0	0
0	0	0	1	0

(1) 原始輸入矩陣

1	1	1	0	1
0	1	0	1	1
0	0	0	0	0
0	0	0	0	0

(2) 將 $M[4,4]$ 改成 0 後，第 4 列全都是 0，刪除第 4 列。

1	1	1	0	1
0	1	0	1	1
0	0	0	0	0
0	0	0	0	0

(3) 第 4 列刪除後，第 3 列是一個邊界線且全都是 0，刪除第 3 列。

1	1	1	0	1
0	1	0	1	1
0	0	0	0	0
0	0	0	0	0

(4) 第 3 列刪除後，矩陣只剩兩列，第 5 行的邊界線是兩個 1，刪除第 5 行。

1	1	1	0	1
0	1	0	0	1
0	0	0	0	0
0	0	0	0	0

(5) 將 $M[2,4]$ 修改成 0 後刪除第 4 行。

1	1	1	0	1
0	1	0	0	1
0	0	0	0	0
0	0	0	0	0

(6) 刪除第一列（三個 1），只剩 1×3 的矩陣。

在上述過程中一共修改了兩個元素，這是所有刪除順序中最少的修改次數，因此，以這個矩陣作為輸入的話，答案是 2。

輸入格式

每筆測試資料的第一行為兩個不超過 25 的正整數 m 和 n ，以下 m 列 n 行是矩陣內容，順序是由上而下，由左至右，矩陣內容為 0 或 1，同一行數字中間以一個空白間隔。

輸出格式

輸出最少的修改次數。

範例一：輸入	範例一：正確輸出
4 5 0 1 0 1 1 1 1 1 0 1 0 0 0 0 0 0 0 0 1 0	2
範例二：輸入	範例二：正確輸出
3 5 0 0 0 1 0 1 0 1 1 1 0 0 0 1 0	1

評分說明

輸入包含若干筆測試資料，對於每一筆測試資料，Python 程式的執行時間限制為 2 秒，其餘程式語言的執行時間限制為 1 秒，依正確通過測資筆數給分。其中：

第 1 子題組 40 分： $m + n \leq 13$ 。

第 2 子題組 60 分：沒有其他限制。