

2023 網際網路程式設計全國大賽

高中組決賽

- 本次比賽共 8 題，含本封面共 26 頁。
- 全部題目的輸入都來自**標準輸入**。輸入中可能包含多組輸入，以題目敘述為主。
- 全部題目的輸出皆輸出到螢幕（**標準輸出**）。

輸出和題目指定的輸出格式必須完全一致，英文字母大小寫不同或有多餘字元皆視為答題錯誤。

- 比賽中上傳之程式碼，使用 C 語言請用 `.c` 為副檔名；使用 C++ 語言則用 `.cpp` 為副檔名。
- 使用 `cin` 輸入速度遠慢於 `scanf` 輸入，若使用需自行承擔 TIMELIMIT 的風險。
- 部分題目有浮點數輸出，會採容許部分誤差的方式進行評測。一般來說「相對或絕對誤差小於 ϵ 皆視為正確」， ϵ 值以題目敘述為主。

舉例來說，假設 $\epsilon = 10^{-6}$ 且 a 是正確答案， b 是你的答案，如果符合 $\frac{|a-b|}{\max(|a|, |b|, 1)} \leq 10^{-6}$ ，就會被評測程式視為正確。

Problem	Problem Name	Time Limit	Memory Limit
A	非確定性義大利麵條醬汁料理	2.5 s	1024 MB
B	NPSC 總統大選	1 s	1024 MB
C	巴乙己放石頭問題	1 s	1024 MB
D	蛋餅愛爬山 2	3 s	1024 MB
E	雞蛋糕	1 s	1024 MB
F	守護魔摩市	3 s	1024 MB
G	猜餅乾	1 s	1024 MB
H	巫醫單車	1 s	1024 MB

2023 網際網路程式設計全國大賽

輸入輸出範例

C 程式範例：

```
1 #include <stdio.h>
2 int main()
3 {
4     int cases;
5     scanf("%d", &cases);
6     for (int i = 0; i < cases; ++i)
7     {
8         long long a, b;
9         scanf("%lld %lld", &a, &b);
10        printf("%lld\n", a + b);
11    }
12    return 0;
13 }
```

C++ 程式範例：

```
1 #include <iostream>
2 int main()
3 {
4     int cases;
5     std::cin >> cases;
6     for (int i = 0; i < cases; ++i)
7     {
8         long long a, b;
9         std::cin >> a >> b;
10        std::cout << a + b << std::endl;
11    }
12    return 0;
13 }
```

A. 非確定性義大利麵條醬汁料理

Problem ID: pasta

還記得舊乙嗎？他現在換了新髮型，變成新乙了。

某天，新乙突然發現他家樓下的便利商店進了一種新的微波食品：非確定性義大利麵條醬汁料理（Nondeterministic Pasta Sauce Cuisine，簡稱 NPSC）。

這種詭譎的微波食品被裝在一個邊長 2023^{1202} 的正方形盒子內，而盒子被一些筆直的非確定性義大利麵條（簡稱麵條）分隔為一些多邊形的格子。每條麵條的兩端都恰好接在盒子上，且不會有整根麵條都貼在盒子邊上的情形發生。而且，為了讓這道料理更加詭譎，所有的麵條都不會互相平行。

每一個被麵條劃分出來的格子內都會填滿一種玄學醬汁，每格的醬汁都不相同。因為這些醬汁太玄學了，在食用這道料理前必須先微波加熱七七四十九秒，否則吃完肚子會發生咻~~~~碰！嘎啦嘎啦！的事情。然而，每根非確定性義大利麵條在加熱之後會獨立的有 $\frac{1}{2}$ 的機率憑空滋~~~~！的消失不見，而只要兩格玄學醬汁中間分隔的麵條不見了，那兩格醬汁就會咕嚕嚕~~~~！的自動混合在一起變成單一一種玄學醬汁。

請問新乙在好好微波加熱這道料理後，期望能吃到多少種玄學醬汁呢？

Input

輸入第一行有一個非負整數 N ，代表非確定性義大利麵條醬汁料理裡有 N 根麵條。接下來有 N 行，第 i 行有四個以空白隔開的整數 a_i, b_i, c_i, d_i ，代表如果將盒子的中心點視為一個平面的原點 $(0, 0)$ 的話，第 i 根麵條會經過 (a_i, b_i) 和 (c_i, d_i) 兩個整數點。

- 所有的輸入都是整數
- $0 \leq N \leq 1000$
- $\forall i \in [1, N], -10^6 \leq a_i, b_i, c_i, d_i \leq 10^6$
- $(a_i, b_i) \neq (c_i, d_i) \forall i \in [1, N]$
- 任兩相異的麵條都不平行

Output

輸出一行包含一個數，代表微波加熱後玄學醬汁的種類數期望值。

由於可以證明答案為有理數 $\frac{P}{Q}$ ，且 Q 不為 998244353 的倍數。為了避免浮點數誤差，請輸出 $PQ^{-1} \pmod{998244353}$ 。其中 Q^{-1} 為 Q 在模 998244353 以下的乘法反元素，也就是 $Q \times Q^{-1} \equiv 1 \pmod{998244353}$ 。

Sample Input 1	Sample Output 1
0	1
Sample Input 2	Sample Output 2
1 0 0 1 1	499122178
Sample Input 3	Sample Output 3
2 0 0 1 1 0 1 1 0	748683267
Sample Input 4	Sample Output 4
4 0 0 3 -2 3 -2 0 3 0 3 -3 -2 -3 -2 0 0	499122181

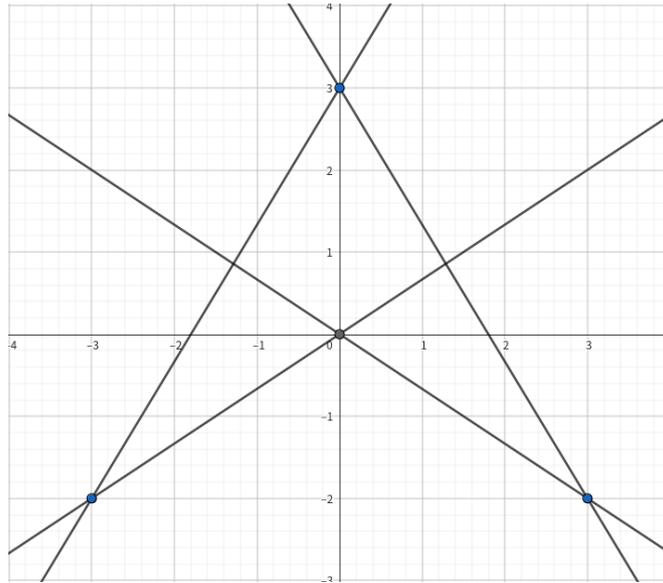
Note

在第一筆範例測試資料中，沒有任何麵條，所以整個盒子只可能有 1 種醬汁。

在第二筆範例測試資料中，有 $\frac{1}{2}$ 的機率麵條不見、剩 1 種醬汁，有 $\frac{1}{2}$ 的機率剩 2 種醬汁，所以答案是 $\frac{1}{2} \cdot 1 + \frac{1}{2} \cdot 2 = \frac{3}{2} \equiv 499122178 \pmod{998244353}$ 。

在第三筆範例測試資料中，有 $\frac{1}{4}$ 的機率麵條都不見、剩 1 種醬汁，有 $\frac{1}{2}$ 的機率一根麵條不見、剩 2 種醬汁，有 $\frac{1}{4}$ 的機率麵條都還在、剩 4 種醬汁，所以答案是 $\frac{1}{4} \cdot 1 + \frac{1}{2} \cdot 2 + \frac{1}{4} \cdot 4 = \frac{9}{4} \equiv 748683267 \pmod{998244353}$ 。

在第四筆範例測試資料的答案是 $\frac{1}{16} \cdot (1 \cdot 1 + 4 \cdot 2 + 6 \cdot 4 + 4 \cdot 7 + 1 \cdot 11) = \frac{72}{16} \equiv 499122181 \pmod{998244353}$ ，麵條分佈如圖所示：



This page is intentionally left blank.

B. NPSC 總統大選

Problem ID: gerrymandering

北太平洋主權聯邦（Nordic Pacifica Sovereign Confederation，簡稱為 NPSC）四年一度的總統大選又要到了，這個國家的選舉制度很特別，採取的是所謂的「間接選舉」制度：選舉委員會會將整個國家劃分為不同的選區，對於每個選區分別去統計每位候選人的得票數，若一名特定候選人在某個選區獲得了過半的票數，則代表該名候選人在該選區中勝出，最後，若存在一名候選人在過半的選區中勝出，則該名候選人成功當選為 NPSC 的總統。

過半的定義：若總共有 M 票，只要票數超過 $\lfloor \frac{M}{2} \rfloor$ 即為過半。

NPSC 由於國土狹長，因此整個國家其實只有一條街，每位居民住在街道上的不同位置上。而因為如此，選區的劃分變成了將這條街道切割成若干個連續的段落，每個段落代表一個選區。每個選區都會統計其中的居民投票結果，以確定該選區的勝出候選人。

在這樣的選舉制度下，「票多的贏，票少的輸」並非永遠正確，如何劃分對自己最有利的選區才是關鍵，而身為選舉委員會主席的你有權力負責這項任務，收到來自各種不同候選人賄賂款項的你想要知道：對於每位候選人，是否存在一種選區劃分方式能夠讓他成功當選為 NPSC 的總統。

Input

輸入第一行有兩個正整數 N, K ，代表一共有 N 位具有投票權居民住在這條街上，且總共有 K 位不同的候選人。

輸入第二行包含 N 個正整數 a_1, a_2, \dots, a_N ，代表每一位居民支持的候選人。

- $1 \leq N, K \leq 2 \cdot 10^5$
- $1 \leq a_i \leq K$

Output

請輸出 K 行，若存在一種選區劃分方式能讓候選人 i 當選，則在第 i 行輸出「Yes」（不含引號），反之則輸出「No」（不含引號）。

Sample Input 1

5 2 1 2 1 2 1

Sample Output 1

Yes No

Sample Input 2

10 3 1 2 2 2 1 1 1 3 1 2

Sample Output 2

Yes Yes No

C. 巴乙己放石頭問題

Problem ID: stone

巴乙己最喜歡收集石頭了，在眾多石頭中，他最喜歡的就是大石頭。今天，他來到全國最著名的珍貴石頭大會（National Precious Stone Conference，簡稱 NPSC），希望能帶走一些有趣的大石頭。

他準備了 N 個容量相同但長相不同的籃子，每個籃子最多能裝兩顆石頭，而且因為籃子寬度很窄，先放入的石頭會在籃子底部，後放入的石頭則會疊在先放入的石頭之上。巴乙己在前往 NPSC 的路上，已經手癢收集了若干顆大石頭並放入其中的一些籃子裡了。

巴乙己到了 NPSC 之後，發現大會準備了 N 種奇特的石頭，每個人每種石頭可以拿走兩顆，但是必須要依序拿第一種到第 N 種石頭。巴乙己有強迫症，他希望同一個籃子內裝的石頭不是同一種，而且一種石頭要拿就要兩顆都拿。到 NPSC 的路上所收集的石頭種類都不相同，而且跟大會準備的石頭種類都不一樣。他知道以他現在籃子的容量，不一定能裝得下 NPSC 送的所有石頭，但他也不想多思考如何好好最大化他能拿的石頭個數，因此他的策略如下：假設他正要拿第 i 種石頭，而且現在還有至少兩個籃子還沒裝滿，則他會選擇其中兩個籃子並將兩顆第 i 種石頭分別放入所選的兩個籃子。如果他找不到兩個還沒裝滿的籃子，或是 N 種石頭都拿完了，他就不會再拿任何石頭。

已知巴乙己到 NPSC 前，第 i 個籃子內已經裝了 a_i 個石頭，請問當他已經拿到不會再拿任何石頭時，籃子的狀態有幾種可能？因為狀態數可能很大，所以請回答這個數量除以 $10^9 + 7$ 的餘數。

我們稱兩個籃子的狀態不同，若存在某一個籃子由頂部到底部所放的石頭種類不同，或是放的石頭個數不同。

Input

第一行輸入一個正整數 N ，意義如題目所述。

第二行輸入 N 個整數 a_1, a_2, \dots, a_N ，其中 a_i 代表到 NPSC 之前，第 i 個籃子裡已經放了 a_i 顆石頭。

- $1 \leq N \leq 10^6$
- $0 \leq a_i \leq 2$

Output

輸出一個整數，代表最後籃子可能的狀態數除以 $10^9 + 7$ 的餘數。

Sample Input 1

4 0 1 0 1	Sample Output 1 19
--------------	-----------------------

Sample Output 1

Sample Input 2

5 0 0 0 0 0	Sample Output 2 2490
----------------	-------------------------

Sample Output 2

D. 蛋餅愛爬山 2

Problem ID: mountain2

蛋餅在全國爬山大賽 (National Pa Shan Contest, 簡稱 NPSC) 獲得冠軍之後, 選擇從爬山競賽退役, 轉型成休閒爬山玩家。事實上, 蛋餅一直都有一個不為人知的技能: 飛天, 由於飛天在全國爬山大賽是被視為作弊的技能, 因此蛋餅一直都沒有機會使用它。現在, 他總算可以盡情的使用這項技能。

蛋餅打算去爬玉大山, 玉大山由 N 座山峰組成, 這 N 座山峰由西到東排成一列, 依序編號為 $1, 2, \dots, N$, 第 i 座山峰的高度是 h_i , 任兩座山峰的高度都不相同。

當蛋餅在第 i 座山峰上面時, 他可以使用飛天技能飛到另一座山峰上, 假設他想要飛到第 j 座山峰, 那必須要滿足:

- 第 j 座山峰比第 i 座山峰矮, 也就是 $h_j < h_i$ 。
- 第 i 座山峰和第 j 座山峰之間的所有山峰, 都比它們兩座山峰還要矮。也就是說, 如果 $i < j$, 那麼要滿足對於所有 $i < k < j$, 都有 $h_k < h_j$; 如果 $i > j$, 要滿足對於所有 $j < k < i$, 都有 $h_k < h_j$ 。

只要滿足上述這些條件, 蛋餅就可以直接從第 i 座山峰飛到第 j 座山峰。

在規劃行程的時候, 蛋餅看著 N 座山峰的高度列表, 想到了 Q 個問題, 第 i 個問題以兩個整數 l_i, r_i 表示, 代表蛋餅想知道「如果蛋餅只能利用飛天技能在第 $l_i, l_i + 1, \dots, r_i$ 座山峰之間移動, 在任選這些山峰的其中一座作為起點的情況下, 蛋餅最多可以使用幾次飛天技能?」

Input

第一行有兩個整數 N, Q , 代表玉大山有幾座山峰和蛋餅想到的問題數量。

第二行有 N 個整數 h_1, h_2, \dots, h_N , 代表玉大山每座山峰的高度。

接下來有 Q 行, 其中的第 i 行包含兩個整數 l_i, r_i , 代表第 i 個問題。

- $1 \leq N, Q \leq 5 \times 10^5$
- $\forall 1 \leq i \leq N, 1 \leq h_i \leq 10^9$

- $\forall i \neq j, h_i \neq h_j$
- $\forall 1 \leq i \leq Q, 1 \leq l_i \leq r_i \leq N$

Output

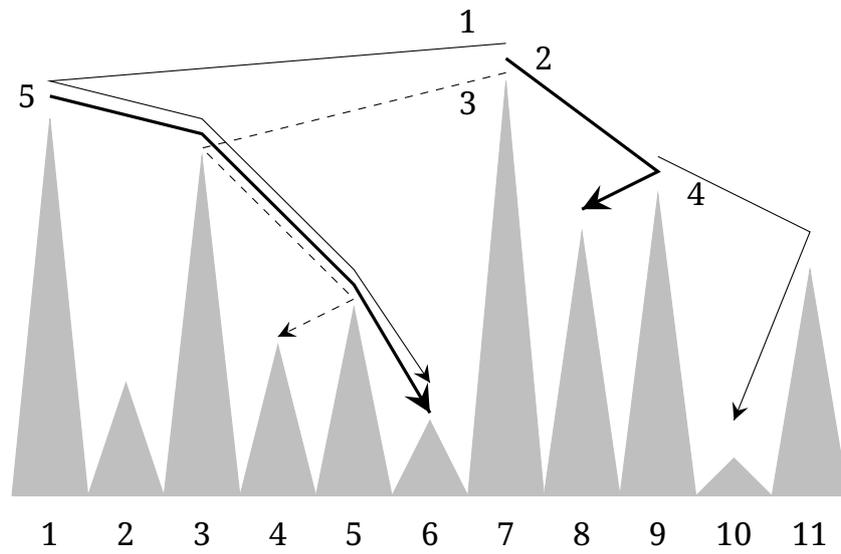
輸出 Q 行，其中第 i 行輸出一個整數，代表第 i 個問題的答案，也就是蛋餅最多可以使用幾次飛天技能。

Sample Input 1	Sample Output 1
11 5	4
10 3 9 4 5 2 11 7 8 1 6	2
1 11	3
4 9	2
3 7	3
8 11	
1 6	

Sample Input 2	Sample Output 2
7 3	1
6 4 2 1 3 5 7	2
4 5	4
3 5	
1 5	

Note

下圖標示數字 i 的路線為範測測試資料 1 中，第 i 個問題的其中一種可以使用最多次飛天技能的移動路線。



This page is intentionally left blank.

E. 雞蛋糕

Problem ID: cake

小冰塊現在正在雞蛋糕的烤爐上，要是再不離開，他就要融化成一灘水了！

烤爐有 N 個橫列，每列從左至右有 N 塊雞蛋糕，現在小冰塊正位於烤爐第一列第一個雞蛋糕的邊緣，他需要通過這些雞蛋糕，從最後一列最後一個雞蛋糕向右離開。

這些雞蛋糕各自有一面是冷的，而另一面則是熱的，要是小冰塊走到熱的那一面上，他便會快速融化、蒸發。因此他在移動的途中只能站在雞蛋糕冷的那一面上。首先，他會站上第一列第一個雞蛋糕，之後的每一步可以往上下左右移動，但是除了從最後一列的最後一個雞蛋糕離開烤爐以外，他都不能往烤爐的邊界以外移動。

目前烤爐上的狀態可能無法讓他通行，幸運的是接下來雞蛋糕店的老闆會不斷的翻動這些雞蛋糕，每次的操作可能會是將一整列的雞蛋糕都翻面，或是將一整行的雞蛋糕都翻面。

現在小冰塊想要知道，有沒有可能經過數次（可能是沒有）翻面後，雞蛋糕上會出現一條可以讓它逃離的路徑？由於冰塊在室溫下也會漸漸融化，他更不可能會冒著被翻下去的風險，所以他不能在雞蛋糕上等待翻面，也就是說，要等待雞蛋糕店的老闆翻完雞蛋糕才能踩上這個雞蛋糕網格。

Input

輸入第一行有一個整數 N ，接下來有 N 行，每行有一個長度為 N 的字串，第 i 行的第 j 個字元代表第 i 列第 j 個雞蛋糕的一開始狀態：如果是 H 代表熱的面朝上，反之是 . 則表示冷的面朝上。

- $1 \leq N \leq 10$

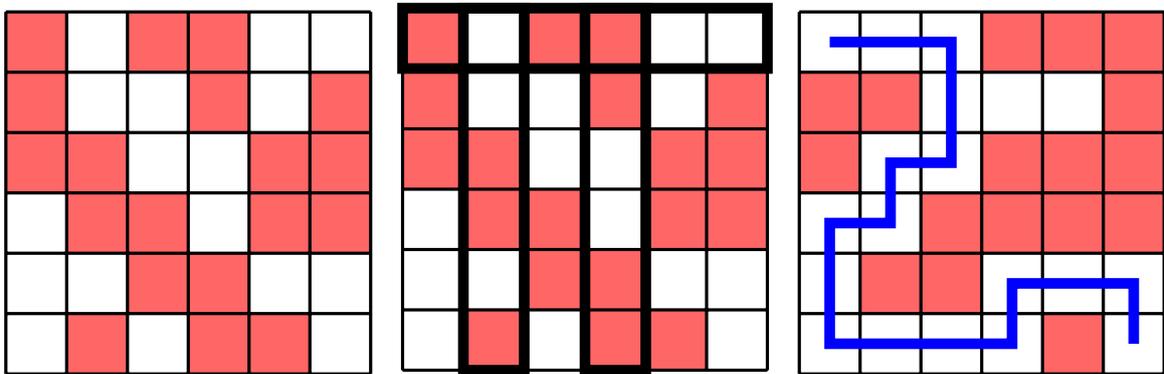
Output

如果小冰塊有可能逃離烤爐，輸出一行 Yes，否則輸出 No。

Sample Input 1	Sample Output 1
<pre>6 H.HH.. H..H.H HH..HH .HH.HH ..HH.. .H.HH.</pre>	<p>Yes</p>

Note

在範例當中，可以透過以下的方式翻轉達成條件。



F. 守護魔摩市

Problem ID: momo

魔摩市，一個長久以來經常被魔族入侵的城鎮。為了生存，居住在魔摩市的魔學家們發明了一種特殊的魔法球，這種魔法球擁有特殊的魔力值，只要將其擺放在魔摩市的中心，就能夠有能力抵禦魔族的入侵。

魔法球一共有四種狀態：靜止、強化、飽滿與衰減。同時魔法球除了魔力值外，還有一種特殊的調節系統：忍耐值。忍耐值是為了防止魔法球在飽滿狀態太久，導致魔法球一直處於過高的魔力值而故障。忍耐值只在飽滿模式時會有用處，每當魔法球從其他狀態變成飽滿狀態，或從飽滿狀態變成其他狀態時，忍耐值都會歸零。

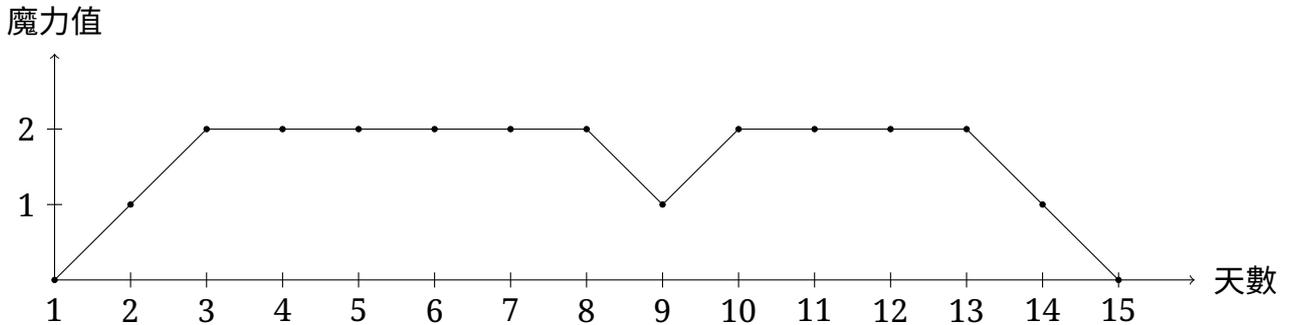
一開始魔法球會處於靜止狀態，每到達新的一天，魔法球會根據目前的狀態、魔力值以及其內部的兩個特殊定值 A, B 來變更魔力值、忍耐值或其狀態，改變的規則如下：

1. 若前一天魔法球處於靜止狀態，則什麼事情都不會發生。
2. 若前一天魔法球處於強化狀態，則魔力值會增加 1。若增加後魔力值到達 A ，則會變成飽滿狀態，並將忍耐值歸零。
3. 若前一天魔法球處於飽滿狀態，則忍耐值會增加 1。若增加後忍耐值到達 B ，則會變成衰減狀態，並將忍耐值歸零。
4. 若前一天魔法球處於衰減狀態，則魔力值會減少 1。若減少後魔力值歸零，則會回到靜止狀態。

由於魔摩市已經是受到嚴重攻擊的城鎮，時常會有討伐魔族的魔法少女來訪城鎮，幫助城鎮抵禦魔族。而她們也很樂意增強魔法球，若當天有魔法少女來訪，她會在晚上的時候根據以下的規則改變魔法球的狀態：

1. 若當前魔法球處於靜止狀態，則她會讓魔法球變成強化狀態。
2. 若當前魔法球處於強化狀態，因為此時的魔法球較危險，為了生命安全她不會做任何事情。
3. 若當前魔法球處於飽滿狀態，她會讓魔法球的忍耐值歸零。
4. 若當前魔法球處於衰減狀態，則若當前魔法球的魔力值為 A ，她會讓魔法球變成飽滿狀態，否則會讓魔法球變成強化狀態。

舉例來說，假設 $A = 2, B = 3$ ，且在第 1, 5, 9 天都有魔法少女來訪，每天的魔力值如下圖所示：



假設第 i 天魔法球的魔力值為 $f(i)$ ，我們定義城鎮的防禦力為 $\sum_{i=1}^{\infty} f(i)$ 。特別注意到晚上行動的魔法少女不會改變魔法球的魔力值，所以一整天的魔力值會是定值。以上面的例子來說，防禦力為 23。

現在魔摩市的魔學家們已經預測了最近 N 天，哪幾天會有魔法少女來訪。然而這畢竟只是預測，結果還是可能跟現實有差距，但可以確定的是某一段連續的日期區間預測結果會是正確的。為此，他們提出了 M 個情況，其中第 i 個情況是除了第 l_i 天到第 r_i 天是預測正確的以外，其他天都沒有魔法少女來訪。對於每一個情況，你能幫幫他們計算城鎮的防禦力是多少嗎？

Input

輸入第一行有四個以空格隔開的正整數 N, M, A, B ，代表預測的天數、情況的數量以及兩個魔法球的特殊定值。

第二行有一個長度為 N 的 01 字串 S ，若 $S_i = 1$ 則代表第 i 天會有魔法少女來訪，否則則沒有。

接下來 M 行，第 i 行會有兩個以空格隔開的正整數 l_i, r_i ，代表第 i 個情況。

- $1 \leq N, M \leq 5 \times 10^5$
- $1 \leq A, B \leq N$
- $|S| = N, S_i \in \{0, 1\}$
- $1 \leq l_i \leq r_i \leq N$

Output

請輸出 M 行，第 i 行請輸出第 i 個情況的防禦力是多少。

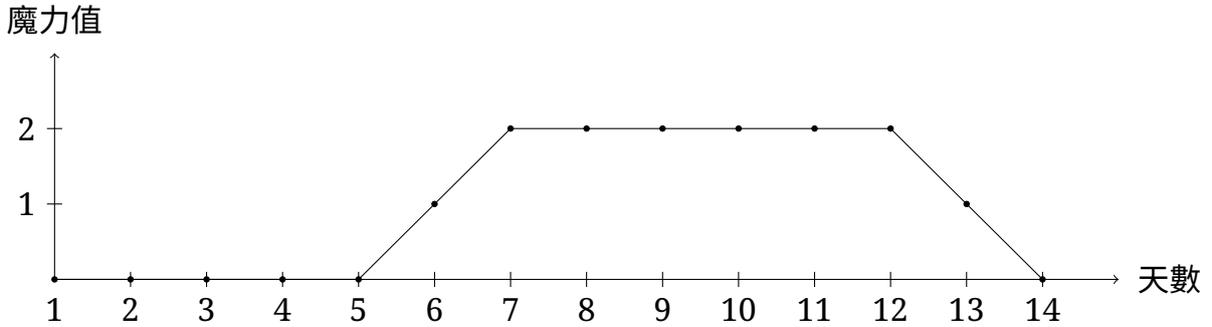
Sample Input 1	Sample Output 1
9 3 2 3 100010001 1 9 2 4 5 9	23 0 14

Sample Input 2	Sample Output 2
10 3 1 2 0100010100 1 10 2 6 3 8	7 6 4

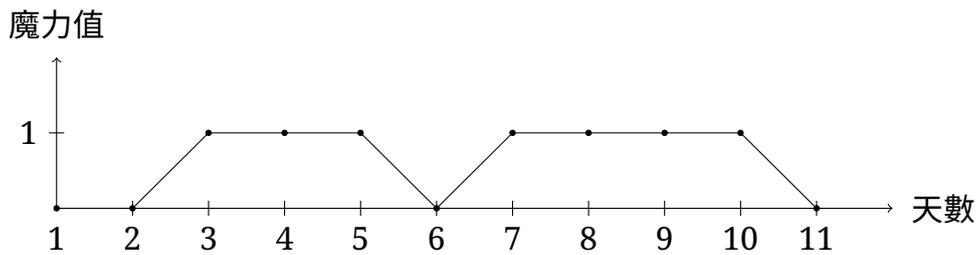
Sample Input 3	Sample Output 3
24 14 3 1 100000100010000010001001 1 24 9 19 8 24 10 22 17 24 4 24 5 13 1 17 1 15 16 23 1 21 7 21 3 18 8 16	57 23 39 31 26 41 15 43 31 15 46 30 27 12

Note

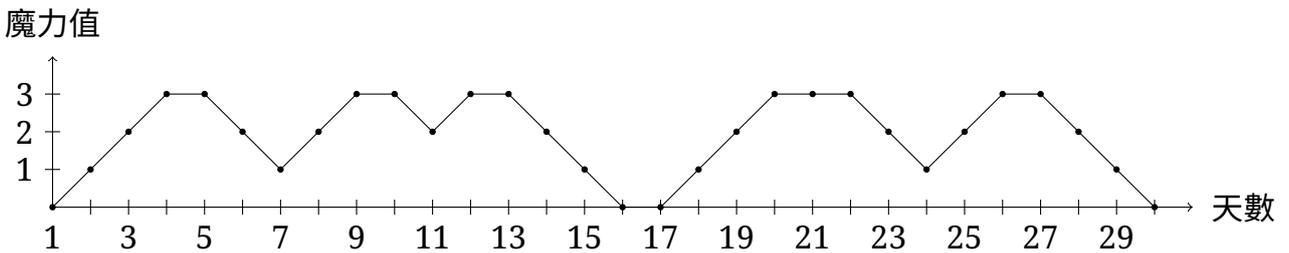
下圖顯示了範例測試資料一的第三個情況下，每天的魔力值變化圖：



下圖顯示了範例測試資料二的第一個情況下，每天的魔力值變化圖：



下圖顯示了範例測試資料三的第一個情況下，每天的魔力值變化圖：



G. 猜餅乾

Problem ID: cookie

本題為互動題。

小董跟小薺是感情很好的一對姐妹，她們想要在學校的文化體育祭上擺設猜餅乾的攤位！

為了擺設攤位，小董跟小薺打算到小橘開設的「黑色薔薇」烘焙坊購買一些手工餅乾。黑色薔薇烘焙坊一共販售 N 種不同口味的餅乾，每一種口味依序以 1 到 N 的正整數編號。在猜餅乾的活動中，她們準備了一個盒子，並且選擇 N 塊餅乾排成一列放入盒中。假設由左到右第 i 塊餅乾的口味為 a_i ，這 N 塊餅乾的口味會以非嚴格遞增的方式排列，也就是說，對於所有的 $1 \leq i < N$ ，皆滿足 $a_i \leq a_{i+1}$ 。在決定好餅乾的順序後，她們會把盒子蓋起來，不讓客人看到裡面的餅乾口味。

當一位客人來到攤位時，猜餅乾的遊戲就正式開始了！首先，小董跟小薺會先告訴客人餅乾的數量 N ，接下來就輪到客人開始猜測盒子裡面的餅乾依序是哪些口味了。每次猜測時，客人必須猜測盒子裡的餅乾依序是哪幾種口味，猜測時這 N 塊餅乾的口味也要以非嚴格遞增的方式排序。也就是說，假設客人猜測由左到右第 i 塊餅乾的口味為 b_i ，則對於所有的 $1 \leq i < N$ ，也必須滿足 $b_i \leq b_{i+1}$ 。

在客人猜完餅乾的種類後，小董跟小薺會在盒子上貼上一些不同顏色的標籤給客人一些提示。首先，對於每一塊餅乾，假設客人猜測的口味與盒子中的口味相同，則她們會在那個位置貼上綠色的標籤。接下來，對於每一種不同的口味，她們會計算該種口味在盒子裡出現的次數 x 以及客人的猜測中該種口味出現的次數 y 。如果 $y > x$ ，則她們會由右向左將前 $y - x$ 塊放著該種口味的餅乾且尚未被貼上綠色標籤的位置貼上灰色標籤。而對於剩下的位置，她們會貼上黃色的標籤。

舉例來說，如果餅乾口味的數量 $N = 7$ ，並且盒子裡放著的餅乾口味 a 為 $[2, 3, 3, 3, 5, 5, 7]$ ，則一些可能的猜測及這個猜測會被貼上的標籤如下圖所示。每一個橫行表示一次客人的猜測，方格中的數字表示猜測的口味，方格的顏色表示被貼上的標籤顏色。

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

小夜是小董和小薺的粉絲，為了吸引她們的注意，她希望在 11 次猜測以內正確得出所有餅乾的口味。你可以寫一支程式幫助她嗎？

互動說明

首先，請從標準輸入 (standard input) 輸入一個正整數 N ，表示餅乾的種類，也就是盒子裡餅乾的數量。

接下來，當你的程式要猜測餅乾的口味時，請輸出一行「? $b_1 b_2 \dots b_N$ 」(不含引號)，依序表示要猜測的每片餅乾的口味。當你輸出完這些數字後，記得要清空 (flush) 標準輸出 (standard output)。

當系統收到你的猜測後，會回覆一行到你的標準輸入。這一行為一個長度為 N 的字串 s ，其中第 i 個字元 s_i 表示第 i 個位置被貼上的標籤顏色。其中，字元「A」表示綠色，「B」表示黃色，「C」表示灰色。

當你確定了所有餅乾的口味時，請輸出一行「! $a_1 a_2 \dots a_N$ 」(不含引號)，依序表示盒子中的每片餅乾的口味。輸出完這行後，你的程式必須立刻結束 (exit)。

你至多只能進行 11 次猜測，確定餅乾口味後的回答不計入這個次數以內。如果你的程式猜測了超過 11 次，或違反了題目敘述中的規則或輸出格式 (例如：猜測的口味沒有滿足非嚴格遞增或不是介於 1 到 N 之間的整數)，你的程式將會被強制中止，並且會得到 WRONG-ANSWER。

- $1 \leq N \leq 1234$
- $1 \leq a_i, b_i \leq N$
- 對於所有的 $1 \leq i < N$ ， $a_i \leq a_{i+1}$ 且 $b_i \leq b_{i+1}$
- $s_i \in \{A, B, C\}$
- 在評測之前，餅乾的順序就已經被決定，不會在過程中被更改

以下兩段程式碼分別為 C 及 C++ 程式的範例，上傳這兩段程式碼並不會讓你得到 CORRECT。範例程式首先讀入 N 的值，並猜測所有餅乾的口味皆為 1，再回答所有餅乾的口味皆為 1 後結束。

```
1 #include <stdio.h>
2 int main()
3 {
4     int N;
5     char s[100];
6     scanf("%d", &N);
7     printf("?");
8     for (int i = 0; i < N; i++)
9         printf(" 1");
10    printf("\n");
11    fflush(stdout);
12    scanf("%s", s);
13    printf("!");
14    for (int i = 0; i < N; i++)
15        printf(" 1");
16    printf("\n");
17    return 0;
18 }
```

```
1 #include <string>
2 #include <iostream>
3 int main()
4 {
5     int N;
6     std::string s;
7     std::cin >> N;
8     std::cout << "?";
9     for (int i = 0; i < N; i++)
10        std::cout << " 1";
11    std::cout << "\n" << std::flush;
12    std::cin >> s;
13    std::cout << "!";
14    for (int i = 0; i < N; i++)
15        std::cout << " 1";
16    std::cout << "\n";
17    return 0;
18 }
```

Sample Input	Sample Output
7	? 1 2 3 3 3 3 7
CBAABCA	? 2 2 3 3 3 4 5
ACAABCB	? 1 1 1 4 4 6 6
CCCCCCC	? 2 3 3 3 5 5 7
AAAAAAA	! 2 3 3 3 5 5 7

H. 巫醫單車

Problem ID: webike

巫醫單車 WeBike 是一個共享單車服務，而這個服務在巫醫巫醫家樓下設有一個可以容納 K 臺單車的站點。一開始，這個站點有 B 輛單車以及 $K - B$ 個空位。

巫醫巫醫用魔法預測到接下來一天會依序發生 N 次事件，每一次事件有可能是：

- 有一個人要還一輛單車。如果這個站點還有空位，那麼還車就會成功，站點的單車數會增加一、空位數會減少一；否則，這個人就會還車失敗。
- 有一個人想要借一輛單車。如果這個站點還有單車，那麼這個人就能成功借到車，站點的空位數會增加一、單車數會減少一；否則，他就會失敗、空手而歸。

巫醫巫醫想要幫助那些會借車、還車失敗的人們。他可以選擇某一次事件，在那個事件發生前施展一次「移車魔法」。移車魔法可以任意把這個站點的單車數量改變成 0 到 K 之間的整數。

現在，巫醫巫醫告訴你了這 N 次事件，請問他要在什麼時間點、把單車的數量變成多少，才能最小化借車、還車失敗的總人數呢？

Input

輸入第一行有三個以空白隔開的整數 N, K, B ，意義如題目敘述所示。第二行有 N 個 0 或 1 字元，第 i 個字元如果是 0 ，則代表第 i 次事件是還車事件。否則，第 i 個字元會是 1 ，代表第 i 次事件是借車事件。

- $1 \leq N \leq 4 \cdot 10^6$
- $1 \leq K \leq 4 \cdot 10^6$
- $1 \leq N \cdot K \leq 4 \cdot 10^6$
- $0 \leq B \leq K$

Output

輸出一行包含三個空白隔開的整數 X, Y, Z ，代表如果巫醫巫醫在第 X 次事件前把站點的單車數量變為 Y 的話，可以最小化借車、還車失敗的總人數為 Z 。如果有多種可行的方法，輸出任意一組皆可。

- $1 \leq X \leq N$
- $0 \leq Y \leq K$
- $0 \leq Z \leq N$

Sample Input 1	Sample Output 1
5 2 1 01100	1 1 0
Sample Input 2	Sample Output 2
7 2 1 0111000	7 0 1
Sample Input 3	Sample Output 3
20 5 0 00000111111111100000	11 5 0
Sample Input 4	Sample Output 4
20 3 2 10001100100110001011	11 1 1