

2020 網際網路程式設計全國大賽

高中組初賽

- 本次比賽共 7 題，含本封面共 18 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**)。
輸出和裁判的答案必須完全一致，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 Time Limit Exceeded 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「相對或絕對誤差不超過 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。

2020 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. 邊緣人

Problem ID: last

有 N 個人要分組，一組 x 個人，編號 1 到 x 的人會分到一組， $(x + 1)$ 到 $2x$ 會分到一組，依此類推。最後可能會有一些人組員人數不足 x ，我們稱這些人為邊緣人。我們定義編號 i 的人的邊緣值 $f(i)$ 為：在 $x = 1, x = 2, \dots, x = N$ 這 N 個情況中，編號 i 的人成為邊緣人的情況總數。

給定 L, R ，請求出 $f(L), f(L + 1), \dots, f(R)$ 。

Input

輸入只有一行，此行有三個由空白隔開的三個整數，分別代表題目的 N, L, R 。

- $1 \leq N \leq 2^{40}$
- $L \leq R \leq N$
- $R - L \leq 3 \cdot 10^5$

Output

輸出只有一行，有 $R - L + 1$ 個由空白隔開的數字，第 i 個數字代表 $f(L + i - 1)$ ，行尾不能有多餘空白並且需有一個換行字元。

Sample Input 1

10 2 10

Sample Output 1

0 0 0 0 0 1 2 4 6

Sample Input 2

100 90 100

Sample Output 2

57 62 64 67 69 71 73 81 82 86 91

This page is intentionally left blank.

B. 握手

Problem ID: handshake

NPSC 有一個傳統，在比賽開始之前，每一位參賽者都必須跟所有其他參賽者握手之後比賽才會開始。亦即，在所有參賽者中任選兩個人，他們都必須跟彼此握過手。

但在 NPSC 中的握手並非只是打招呼那麼簡單。每一位參賽者都有一個**電力值** d ，而當兩個人握手時，整個會場的**氣氛值**會產生變化。考慮兩位參賽者小 T 與小 D，假設小 T 的電力值是 d_T ，小 D 的電力值是 d_D ，且 $d_T > d_D$ 。則當兩個人握手時，小 T 會因為覺得自己可以電小 D 而感到開心，使整個會場的氣氛值提高 d_T ，但小 D 會覺得自己被小 T 電而感到沮喪，使整個會場的氣氛值下降 d_D 。若兩個人的電力值相同，則他們握手不會造成氣氛值的變化。

以往握手總是可以正常進行，但是今年因為邪惡的病毒造成疫情，參賽者必須保持社交距離而不能彼此握手。但是 NPSC 主辦方仍然想要知道如果今年的參賽者都互相握過手之後，現場的**氣氛值**會變化多少？

Input

輸入的共有 2 行，第一行是一個正整數 n ，代表今年的參賽者人數。第二行有 n 個用空白分隔的非負整數，其中第 i ($1 \leq i \leq n$) 個值 d_i 代表第 i 位參賽者的電力值。

- $1 \leq n \leq 2 \cdot 10^5$
- $0 \leq d_i \leq 10^5$ ，對於所有 $1 \leq i \leq n$

Output

請輸出一個整數，代表如果每位參賽者都彼此握過手之後，現場**氣氛值**的變化。

Sample Input 1	Sample Output 1
4 4 2 0 3	13

Sample Input 2

6 12 7 5 20 7 1

Sample Output 2

116

Sample Input 3

15 10 13 19 2 43 28 20 36 28 23 28 10 48 31 46

Sample Output 3

1704

C. 俄羅斯方塊

Problem ID: tetris

NPSC 遊戲公司最近開發了一款新的俄羅斯方塊遊戲。具體來說，這個遊戲由一個 $N \times M$ 的網格組成。為了方便說明，以下把每個 1×1 格子稱為「單元」，最左上角的單元座標為 $(0, 0)$ ，最右下角的單元座標為 $(N - 1, M - 1)$ 。在一開始，這個網格中會有任意數量的單元已經被先前落下的俄羅斯方塊占用。為了簡單起見，在本題當中不考慮方塊的消去，也就是有可能一整列的格子都是已經被占用的單元。

遊戲進行過程中，會有一個俄羅斯方塊開始落下。不同於傳統的俄羅斯方塊遊戲，在這個遊戲中每一個落下的俄羅斯方塊可能由 1 到 11 個單元組成，而這些單元中其中一個是「中心單元」。組成俄羅斯方塊的所有單元必定四方位連通，也就是說從該方塊的任一個單元開始，可以在該方塊內的單元中上下左右移動而抵達該方塊的另一個單元。俄羅斯方塊落下的規則如下：

- 以下以「發生碰撞」代表正在落下的方塊中任意一個單元與已經被占用的單元重疊、或超出遊戲網格的現象。
- 每隔一段時間，方塊會「自由落下」一格的距離，也就是把整個俄羅斯方塊往下平移一格。如果在平移之後發生碰撞，那麼方塊將被固定在**平移前的位置**，且整個落下的過程結束。
- 在落下的過程結束之前，可以在任何時間（不與自由落下同時）以任意順序進行任意次數的「平移」與「旋轉」操作，前提是每次操作之後都不能發生碰撞：
 - 「平移」有向左與向右兩種選項，分別是把整個俄羅斯方塊往左或往右平移一格。
 - 「旋轉」有逆時鐘或順時鐘兩種選項，分別是把整個俄羅斯方塊以「中心單元」為中心逆時鐘或順時鐘旋轉 90 度。

請你寫一個程式，輸入一個盤面，判斷一個正在落下的俄羅斯方塊在落下過程結束之後所有可能的位置與方向。

Input

第一行有兩個以空白隔開的正整數 N, M ，代表遊戲盤面的高度和寬度。

接下來有 N 行，每行有只由 `.oX` 四種字元、長度為 M 的字串，代表初始的盤面。四種字元代表的意義如下：

- X：正在落下的方塊的中心單元。
 - x：正在落下的方塊的非中心單元。
 - o：一開始已經被佔用的單元。
 - .：未被佔用的單元。
- $N, M \leq 1000$

Output

請輸出 X 行，代表該正在落下的方塊在落下過程結束後有 X 種可能的位置與方向。

每行請輸出三個以空白隔開的非負整數 x, y, d ，代表一個可能的位置與方向，其中 x, y 代表該方塊**中心單元**最後位置的座標為 (x, y) ，而 d 代表方塊最後的方向，0, 1, 2, 3 分別代表方塊一開始、逆時鐘旋轉 90 度、旋轉 180 度與順時鐘旋轉 90 度的方向。**注意就算旋轉之後方塊的外型與旋轉前相同，仍要視為有旋轉過。**

輸出必須按照 (x, y, d) 的遞增順序排序。

Sample Input 1

9 5	5 2 0
..X..	5 2 3
.oXo.	5 3 0
o.X.o	6 2 2
o.Xo.	6 3 2
.O...	
....o	
O...o	
O....	
OOOOO	

Sample Output 1

Sample Input 2

3 3	1 1 0
xxx	1 1 1
xXx	1 1 2
xxx	1 1 3

Sample Output 2

D. 隨機並查集

Problem ID: unionfind

你知道並查集 (Disjoint-set Union) 嗎？並查集是一種可以處理「快速合併兩個集合」並且詢問「某個元素所在的集合」這樣操作的資料結構。具體來說，對於 N 個由 1 到 N 形成的集合

$$S = \{\{1\}, \{2\}, \dots, \{N\}\}$$

並查集可以支援下列兩種操作：

1. 合併：指定 x 與 y ，合併元素 x 與元素 y 所屬的集合。
2. 查詢：指定 x 與 y ，查詢元素 x 與元素 y 是否同屬一個集合。

小 B 最近在演算法課程上學到了這個資料結構。專精隨機演算法的他，馬上開始好奇這個資料結構的「隨機」所需的執行週期 (cycle) 數。具體來說，在程式開始執行前，每個元素都在不同的集合。程式會花一個週期隨機選取 $1 \leq x < y \leq N$ 並判斷 x 與 y 是否位於同一個集合。若 x 與 y 位於同個集合，則程式不做任何額外的事。若 x 與 y 分別位於集合 A 與集合 B ，小 B 的程式會額外花 $f(|A|, |B|)$ 個週期將集合 A 與集合 B 合併。這個過程會持續到所有元素都位於同一個集合為止。

以下的虛擬碼模擬了小 B 的程式：

Algorithm 1 Stochastic Union-Find

```
1: procedure Emulate
2:    $cycle \leftarrow 0$ 
3:    $S \leftarrow \{\{1\}, \{2\}, \dots, \{N\}\}$ 
4:   while  $|S| > 1$  do
5:      $cycle \leftarrow cycle + 1$ 
6:     randomly choose two distinct elements  $x$  and  $y$ 
7:     if  $x$  and  $y$  belongs to sets  $A \neq B$  then
8:        $cycle \leftarrow cycle + f(|A|, |B|)$ 
9:       merge  $A$  and  $B$ 
return  $cycle$ 
```

小 B 想知道他的程式執行所需週期數的期望值，請你設計另一個程式幫幫他吧！

Input

輸入第一行有一個正整數 N ，代表元素的個數。接著有 N 行，每行有 N 個非負整數，第 i 行的第 j 個代表 $f(i, j)$ 。保證 $f(i, j) = f(j, i)$ 且對於所有 $i + j > N$ ，不可能有合併大小 i 與大小 j 的集合的操作，因此方便起見 $f(i, j) = 0$ 。

- $1 \leq N \leq 40$
- $0 \leq f(i, j) \leq 10^9$

Output

輸出所需週期數的期望值。可以證明答案為有理數 $\frac{P}{Q}$ ，為了避免浮點數誤差，請輸出 $PQ^{-1} \pmod{10^9 + 7}$ 。其中 Q^{-1} 為 Q 在模 $10^9 + 7$ 以下的乘法反元素，也就是 $Q \times Q^{-1} \equiv 1 \pmod{10^9 + 7}$ 。

Note

範例測試資料一中，總共只有一個集合，因此程式在一開始就結束，並不花費任何週期。

範例測試資料二中，一開始有兩個集合。第一個週期有 100% 的機率選到元素 1 與 2，並額外花一週期將它們合併。總共需要兩個週期。

範例測試資料三中，期望所需週期數為 $\frac{P}{Q} = \frac{5}{2}$ ，可以自行驗證 $500000006 \times 2 \equiv 5 \pmod{10^9 + 7}$ 。

數字 x 在模 $10^9 + 7$ 以下的乘法反元素為 $y \equiv x^{10^9+5} \pmod{10^9 + 7}$ 。

Sample Input 1	Sample Output 1
1 0	0

Sample Input 2	Sample Output 2
2 1 0 0 0	2
Sample Input 3	Sample Output 3
3 0 0 0 0 0 0 0 0 0	500000006
Sample Input 4	Sample Output 4
10 7 7 3 4 2 7 6 7 7 0 7 5 4 8 10 4 7 6 0 0 3 4 5 1 4 4 3 0 0 0 4 8 1 6 8 3 0 0 0 0 2 10 4 8 4 0 0 0 0 0 7 4 4 3 0 0 0 0 0 0 6 7 3 0 0 0 0 0 0 0 7 6 0 0 0 0 0 0 0 0 7 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	979356198

This page is intentionally left blank.

E. 貓貓排序

Problem ID: sorting

殿王是個天才兒童，他在一個月大的時候就學會數數、六個月大的時候就學會乘法跟除法、一歲時學會寫程式、一歲又六個月時養了可愛的拉不拉多、一歲又十個月時養了可愛的貓咪、兩歲時發明了「吃餅乾」的遊戲，現在要講的是殿王三歲三個月大時的故事。

這天殿王在跟他養的貓咪們玩的時候想到了一個有趣的事情。

如果我們先對所有貓咪排成一列，根據他的毛色、個性等等數值算出對每隻貓分別的好感度，再將這些貓咪按照好感度由小到大排序，那麼這個殿王就稱這個貓咪序列是可愛貓咪序列。

然而對貓咪排序並不是一件簡單的事情。為了怕傷害到那些貓咪，在排序貓咪時唯二能做的操作只有選擇連續的兩隻貓咪或是三隻貓咪，並將牠們的順序反轉。雖然無論哪種操作都不是件省力的事情，但是因為殿王的貓咪特別喜歡被第二種操作（也就是選擇連續的三隻並反轉的操作），所以殿王特別喜歡第二種操作，以致於在排序這些貓咪時殿王都會盡可能的最小化第一種操作的操作次數。

這天，殿王突發奇想的開始好奇，如果他只想排序某個連續區間的貓咪，那麼他最少所需要的第一種操作次數是多少。當然殿王是不會被這個問題難倒的，但是為了保險起見，還是請你寫一個程式幫殿王驗證一下他的答案是不是正確的吧。

Input

輸入的第一行是一個正整數 N ，代表總共有 N 隻貓咪。

第二行有 N 個正整數 a_1, a_2, \dots, a_n ，代表 N 隻貓咪依序的好感度。

第三行有一個正整數 Q ，代表殿王好奇的連續區間總數。

接下來的 Q 行，每行會有兩個正整數 l, r ，代表殿王想要請你幫忙驗算如果想要排序好第 l 隻貓咪到第 r 隻貓咪，最少所需要的第一種操作次數是多少。

- $1 \leq N, Q \leq 40000$
- $1 \leq a_i \leq 10^9$ ，對所有合法的 i
- $1 \leq l \leq r \leq N$ ，對每一筆殿王的詢問

Output

對於每一個國王問題，請輸出對應的答案。

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

F. 兔田建設

Problem ID: usadaconstruction

こんぺここんぺこー

西元 20xx 年，有個知名的競賽程式 vtuber 團體，致力於提升自己在各大 online judge 的 Elo rating，史稱 EloLive。

其中，又有擅長構造題的一幫人，被稱做「兔田建設」(Usada Construction)。

今天，兔田建設收到一個委託，要在一片草原上構造一間南瓜屋。這片草原非常特殊，可以被想像成一個整數序列，其中每個數值代表着一個位置草的多寡。

他們希望這間寬度為 k 的南瓜屋被建造在一個區間 $[l, r]$ 中。除此之外，他們希望南瓜屋建造的地方恰好有 c 塊草地的數值在 v 以下，如此這間南瓜屋的生草值才會恰到好處。不過，因為兔田社長總是出爾反爾的（而且有時候會分不清楚東南西北），所以他們決定了一條清單，清單上有 q 種可能的方案 (v_i, l_i, r_i, c_i) 。而他們想要知道，對於每一種方案，是否能夠建構出符合條件的南瓜屋。

正式地說，給定一個長度為 n 的整數序列 a_1, a_2, \dots, a_n 以及一個固定寬度 k ，你需要回答 q 筆詢問，每一筆詢問 (v_i, l_i, r_i, c_i) 代表是否能在序列的 $[l_i, r_i]$ 區間內找到一個長度為 k 的連續區間，使其恰好有 c_i 個不大於 v_i 的值。

Input

輸入的第一行包含兩個整數 n 和 k ，表示草原的長度以及南瓜屋的寬度。

第二行有 n 個整數 a_1, a_2, \dots, a_n ，表示代表草原的整數序列。

第三行有一個整數 q ，表示詢問的次數。

接下來有 q 行，每一行表示一筆詢問。每筆詢問包含四個整數 (v_i, l_i, r_i, c_i) ，詢問是否能在 $[l_i, r_i]$ 區間內建造出恰好蓋住 c_i 塊不大於 v_i 的草地的南瓜屋。

- $1 \leq n, q, k \leq 10^6$
- $1 \leq a_i \leq 10^9$

- $1 \leq v_i \leq n$
- $1 \leq l_i \leq r_i \leq n$
- $0 \leq c_i \leq k$

Output

對於每筆詢問，請輸出 1 或 0，分別代表能或不能建造出符合條件的南瓜屋。

Sample Input 1	Sample Output 1
3 1 7 7 2 2 4 3 3 0 2 1 3 0	0 1

Sample Input 2	Sample Output 2
4 2 12 9 7 11 2 17 2 2 2 5 1 2 1	0 0

Sample Input 3	Sample Output 3
10 3 4 17 19 20 22 24 26 11 16 13 5 15 6 10 3 30 1 7 0 22 8 9 3 23 3 10 2 18 1 8 1	0 0 0 1 1

G. 排隊

Problem ID: queue

這天，一款全新的妹妹虛擬實境遊戲即將發售，身為這類遊戲的愛好者——小 Y 為了搶先體驗，他現在身處在一個很長的隊伍中，漫長的等待時間使他非常無聊，所以他決定開始觀察隊伍的排隊現象。

小 Y 觀察到隊伍內總共有 N 個人，一開始隊伍是在筆直的數線上的，第 i 個人的座標為 i 。小 Y 也知道自己是這隻隊伍的第 $Q + 1$ 個人，所以他很在意前 Q 個人依序買到遊戲後，隊伍會發生什麼事。

因此，小 Y 接下來會觀察 Q 次事件，第 i 次事件排在隊伍最前面的人，也就是第 i 個人，就會買到遊戲，並離開隊伍，自此這個人的位置便會出現空格，此時隊伍可能會開始移動。

不過常常會有討厭的人待在原地不動（像是滑手機之類的），導致後面的人會因為察覺不到隊伍有移動而無法前進，小 Y 發現對於第 i 次事件，有往前移動的人是還待在原隊伍的前 k_i 位（也就是第 $i + 1 \sim i + k_i$ 個人），這 k_i 位會不受影響地依序直接移動到前 k_i 個位置，也就是座標 $1 \sim k_i$ 。

小 Y 已經把所有事件的 k_i 都紀錄下來了，你可以告訴他，對於每次事件，隊伍內所有人（包含該次事件買到遊戲的人）移動的距離總和是多少呢？當然，我們假設每次事件只有買到遊戲的人和待在原隊伍的前 k_i 個人會做移動，且任何人都只會在事件發生時移動。

Input

輸入的第一行有兩個正整數 N, Q ，代表隊伍的人數、小 Y 前面排了多少人。

第二行有 Q 個以空格分開的整數 k_1, k_2, \dots, k_Q ，代表每次事件，往前移動的人是還待在原隊伍的前 k_i 位，若 $k_i = 0$ 代表此次事件除了買遊戲的人以外，沒有其他的人移動。

- $1 \leq N \leq 10^9$
- $1 \leq Q \leq \min(N - 1, 10^6)$
- $0 \leq k_i \leq N - i$

Output

輸出只有一行 Q 個數字 d_1, d_2, \dots, d_Q ，代表第 i 次事件中，所有人移動的距離總和是 d_i 。數字之間以單一空格隔開，最後一個數字後面必須是一個單一的換行字元。

Sample Input 1

10 5 3 4 2 5 3	Sample Output 1
-------------------	-----------------

Sample Output 1

3 6 2 15 3

Sample Input 2

8 6 3 1 2 1 0 0	Sample Output 2
--------------------	-----------------

Sample Output 2

3 1 5 1 0 5

Sample Input 3

1000000000 7 100 1000 10000 8787 888777 98765 123456

Sample Output 3

100 1901 28002 8787 4405105 98765 148148
--