

注意事項

1. 本次競賽共有 6 題，每題滿分皆為 100 分，共 600 分。本次競賽的總得分為各題的得分總和。
計算排名時, 以總得分由高至低排序為主，若有同分的狀況，則以最後一次分數變動的時間點為基準, 較早者名次較高。
2. 本次競賽使用 CMS 系統。CMS 系統說明網址：<https://reurl.cc/EZ7qxa>。
3. 本次競賽每題皆由數個子任務(子題) 構成，每個子任務中有數筆測試資料，需完全答對某一子任務中的所有測試資料才能獲得該子任務的分數。單次上傳得分為所有子任務得分的總和，該題得分為每個子任務得到的最大得分的總和。
例如，假設某一題有兩個子任務。某人傳了兩份程式，第一份程式只成功解決第一個子任務，第二份程式只成功解決第二個子任務，則這個人得到的分數為兩個子任務的分數和。
4. 本次競賽不得參考任何線上資料、隨身碟等電子式的資料，系統附的「STL 標準樣板函數庫」不在此限。可以參考紙本資料。
在比賽過程中禁止瀏覽本系統、「STL 標準樣板函數庫」以外之網頁。請勿嘗試以任何方法破壞本系統。
5. 若需要裝水、上廁所等，請通知監考人員。
6. **本次競賽不提供即時計分板。**
7. 作答的小技巧
 - 題目要看清楚，重要的條件不要漏看。
 - 題目有很多很簡單的子任務，當想不到整體題目怎麼解決時，可以先想子任務拿到部分分數。
 - 做好時間分配，適當的把整場比賽的時間分配到每一題上面。
 - 對於使用 C++ 輸入輸出 (cin、cout) 的參賽者，由於題目輸入較多，為改善輸入輸出之效能，請在 `int main()` 的第一行插入以下程式碼，並且請勿同時使用 C 輸入輸出 (scanf、printf、puts)。
`ios_base::sync_with_stdio(0); cin.tie(0);`

祝比賽順利！

第一題：數字反轉 (Reverse2)

問題敘述

給定一個以十進位寫成的正整數 n ，定義 $f(n)$ 為將 n 所有位數反轉之後得到的正整數（若有前導 0 則會移除）。例如 $f(372) = 273$, $f(9921) = 1299$, $f(2300) = 32$, $f(40000) = 4$ 。

給定兩個正整數 l, r ，請問有多少個介於 l 和 r 之間（含 l 與 r ）的正整數 x 滿足 $f(x) > x$ ？

輸入格式

多筆輸入。輸入第一行含有一個正整數 t ，代表測試資料的數量。

每筆測試資料只有一行，含有兩個正整數 l, r 。

輸出格式

對於每筆測試資料輸出一個整數於一行，代表有多少個介於 l 和 r 之間的正整數 x 滿足 $f(x) > x$ 。

測資限制

- $1 \leq t \leq 10$
- $1 \leq l \leq r \leq 10^5$

輸入範例 1

```
5
29 29
37273 37273
100000 100000
32024 79201
1 100000
```

輸出範例 1

```
1
0
0
20708
44451
```

評分說明

本題共有 3 組測試題組，條件限制如下所示。每一組可有一或多筆測試資料，該組所有測試資料皆需答對才會獲得該組分數。

子任務	分數	額外輸入限制
1	30	$1 \leq l \leq r \leq 9$ 。
2	30	$1 \leq l \leq r \leq 99$ 。
3	40	無額外限制。

第二題：無理數乘冪 (Power2)

問題敘述

給定兩個非負整數 a, b 與兩個正整數 c, k ，其中 c 不是完全平方數（也就是說，不存在一個整數 r 滿足 $r^2 = c$ ）。

已知在以上條件下，存在唯一一個非負整數對 (p, q) 滿足 $(a + b\sqrt{c})^k = p + q\sqrt{c}$ ，請求出 p, q 的值分別是多少。由於答案可能非常大，請輸出 p, q 兩數的值除以 $10^9 + 7$ 的餘數。

輸入格式

多筆輸入。輸入第一行含有一個正整數 t ，代表測試資料的數量。

每筆測試資料只有一行，含有四個整數 a, b, c, k 。

輸出格式

對於每筆測試資料輸出兩個整數於一行，分別代表 p, q 兩數的值除以 $10^9 + 7$ 的餘數。

測資限制

- $1 \leq t \leq 10^5$
- $0 \leq a, b \leq 10^9$
- $2 \leq c \leq 10^9$ ，且 c 不是完全平方數
- $1 \leq k \leq 10^9$

輸入範例 1

```
10
2 0 2 3
3 0 7 8
9 0 12 100
1000000000 0 1000000000 1000000000
1 1 2 1
1 1 2 3
4 2 5 2
1 1 2 25
14874 4009242 2890142 493967501
343232518 188491321 996002494 773103994
```

輸出範例 1

```
8 0
6561 0
136318165 0
312556845 0
1 1
7 5
36 16
855077834 311738114
88820533 133092081
360520549 629521575
```

備註

一些輸入範例的解釋：

- 第 1 筆輸入中 $(2 + 0\sqrt{2})^3 = 2^3 = 8 = 8 + 0\sqrt{2}$ ，因此輸出 **8 0**。
- 第 2 筆輸入中 $(3 + 0\sqrt{7})^8 = 3^8 = 6561 = 6561 + 0\sqrt{7}$ ，因此輸出 **6561 0**。
- 第 6 筆輸入中 $(1 + \sqrt{2})^3 = 7 + 5\sqrt{2}$ ，因此輸出 **7 5**。
- 第 8 筆輸入中 $(1 + \sqrt{2})^{25} = 1855077841 + 1311738121\sqrt{2}$ ，但是答案要除以 $10^9 + 7$ 的餘數，因此輸出 **855077834 311738114**。

評分說明

本題共有 5 組測試題組，條件限制如下所示。每一組可有一或多筆測試資料，該組所有測試資料皆需答對才會獲得該組分數。

子任務	分數	額外輸入限制
1	9	$b = 0, k \leq 100$ 且 $a^k < 10^9 + 7$ 。
2	17	$b = 0, k \leq 100$ 。
3	28	$b = 0$ 。
4	20	$k \leq 100$ 。
5	26	無額外限制。

第三題：遊戲升等 (Level-up)

問題敘述

Carol 在疫情期間下載了一個手機遊戲來消遣時光。

這款手機遊戲包含一個由玩家操控的主角，主角共有 k 個等級，分別為等級 1 至等級 k 。玩家每天都可以登入這個遊戲，並且可以自由選擇是否要升級角色，但是**每天最多只能升一等**。

玩家每天登入時會有登入獎勵，在玩家**做任何事情前領取**（包含升等）。登入獎勵為 1 至 10^4 枚不等的金幣，由玩家登入的日期與主角等級決定，每天只能領取一次。

Carol 會在接下來的 n 天每天都玩這個遊戲。Carol 的主角一開始是等級 1（可以領取第 1 天等級 1 的登入獎勵），而 Carol 想要在第 n 天**登入時**就已經是等級 k （並領取第 n 天等級 k 的登入獎勵）。熱心的網友已經幫忙算出接下來 n 天在每個等級時登入會獲得多少金幣，而 Carol 想知道這 n 天只靠登入獎勵最多可以獲得多少金幣。你能幫忙她嗎？

輸入格式

單筆輸入。第一行有兩個正整數 n, k ，代表遊戲遊玩天數與角色等級數。

接下來 n 行，每行有 k 個正整數 $coin_{i,1}, coin_{i,2}, \dots, coin_{i,k}$ ， $coin_{i,j}$ 代表在第 i 天等級 j 的登入獎勵金幣數。

輸出格式

輸出一個整數，代表 Carol 這 n 天靠登入獎勵最多可以獲得多少金幣。

測資限制

- $1 \leq n \leq 10^5$
- $1 \leq k \leq 5$
- $k \leq n$
- $1 \leq coin_{i,j} \leq 10^4$

輸入範例 1

```
5 3
1 4 3
5 2 6
2 3 8
```

3 6 3
9 9 1

輸出範例 1

16

輸入範例 2

6 1
10000
10000
10000
10000
10000
10000
10000

輸出範例 2

60000

輸入範例 3

3 3
1 100 100
100 1 100
100 100 1

輸出範例 3

3

輸入範例 4

6 2
1 1
1 100
34 1
35 1

34 1

1 1

輸出範例 4

106

備註

第一筆範例輸入的其中一種可以拿到最多金幣的方法如下：

- 第一天：角色等級為 1，領取 1 枚金幣，不升等
- 第二天：角色等級為 1，領取 5 枚金幣，升到等級 2
- 第三天：角色等級為 2，領取 3 枚金幣，不升等
- 第四天：角色等級為 2，領取 6 枚金幣，升到等級 3
- 第五天：角色等級為 3，領取 1 枚金幣，已是最高等級。

故答案是 $1 + 5 + 3 + 6 + 1 = 16$ 枚金幣。

評分說明

本題共有 6 組測試題組，條件限制如下所示。每一組可有一或多筆測試資料，該組所有測試資料皆需答對才會獲得該組分數。

子任務	分數	額外輸入限制
1	10	$k = 1$ 。
2	13	$k = 2, n \leq 1000$ 。
3	17	$k = 2$ 。
4	11	$n \leq 30$ 。
5	23	$n \leq 1000$ 。
6	26	無額外限制。

第四題：黑白棋 (Token)

問題敘述

Dan 在白紙上畫了 n 個排成一排的格子，由左至右的編號分別為 1 至 n 。

每個格子上面都寫有一個正整數，由左至右分別為 a_1, a_2, \dots, a_n 。其中第一格寫著 1 (即 $a_1 = 1$)，且對於所有介於 2 到 n 之間的正整數 i 而言都有 $a_i < i$ 。

現在 Dan 找來了一枚黑棋與一枚白棋，想要玩一個無聊的遊戲。黑棋與白棋一開始時分別在編號為 u 與 v 的格子上。在每分鐘，假設現在黑棋與白棋分別在編號為 x 與 y 的格子上，那 Dan 就會**同時**把黑棋從 x 移動到 a_x ，把白棋從 y 移動到 a_y 。Dan 想要請教你最少需要幾分鐘才能讓兩枚棋子相遇。

由於 Dan 可能會玩這個遊戲很多次，因此他可能會詢問你很多次。

輸入格式

單筆輸入。第一行有兩個正整數 n, q ，代表有幾個格子以及詢問次數。

第二行有 n 個正整數 a_1, a_2, \dots, a_n ，代表每個格子上面分別寫的數字。

接下來 q 行，每行有兩個正整數 u, v ，代表一筆詢問：一開始時黑棋在 u ，白棋在 v 。

輸出格式

輸出 q 行。對於每筆詢問輸出一個整數於一行，代表黑棋與白棋最少需要幾分鐘才能相遇。如果黑棋與白棋一開始就重合了，輸出 0。

可以證明在給定條件下兩枚棋子必定會在有限時間內相遇。

測資限制

- $1 \leq n, q \leq 3 \times 10^5$
- $a_1 = 1$
- 對於所有介於 2 至 n 的正整數 i ，都滿足 $1 \leq a_i < i$ 。
- $1 \leq u, v \leq n$

輸入範例 1

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

3 4
1 1

輸出範例 1

3
1
0

輸入範例 2

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

輸出範例 2

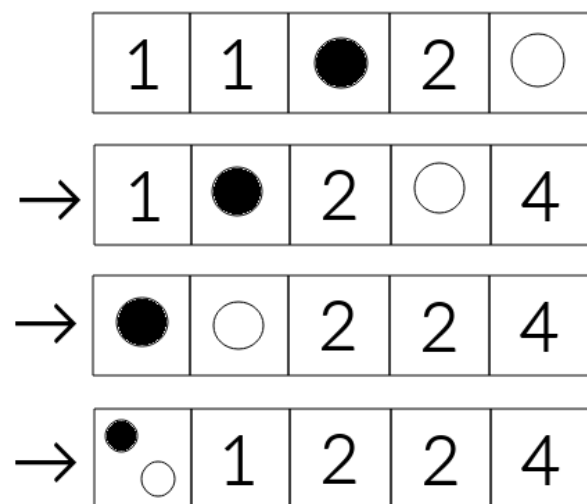
3
0
8
7

備註

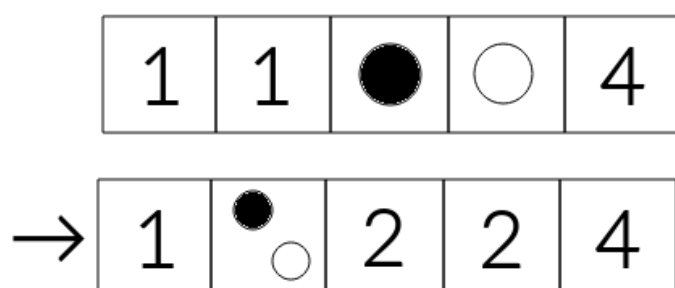
子任務 3 的額外輸入限制為：若一開始有一枚棋子在編號 i 的格子，可以求出最少需要移動幾次才能從 i 號格子移到 1 號格子（一次移動指的是從 x 移到 a_x ），稱這個值為 d_i 。則對於每筆詢問都有 $d_u \neq d_v$ 。

第一筆範例輸入每筆詢問的圖示如下：

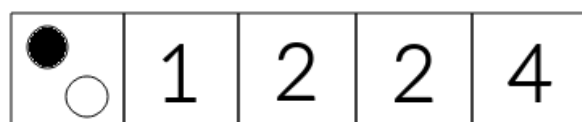
第一筆詢問：



第二筆詢問：



第三筆詢問：



評分說明

本題共有 4 組測試題組，條件限制如下所示。每一組可有一或多筆測試資料，該組所有測試資料皆需答對才會獲得該組分數。

子任務	分數	額外輸入限制
1	14	$n, q \leq 1000$ 。
2	25	$u \neq v$ ，且對於所有介於 2 到 n 之間的 i 都有 $a_i = i - 1$ 。
3	21	見備註。
4	40	無額外限制。

第五題：平方投票法 (Voting)

問題敘述

平方投票法 (Quadratic Voting) 是由法學院教授 Eric Posner 與經濟學家 Glen Weyl 發明的一種投票方式。相較於一般投票提供每位選民相同票數，平方投票法提供每位選民相同的點數，投票時若想要給某一提案 x 票，則需要付出 x^2 的點數。例如，假設有五個提案，某位選民在五個提案分別想要投 2, 0, 3, 1, 3 票，則他需要付出 $2^2 + 0^2 + 3^2 + 1^2 + 3^2 = 23$ 點。

根據 Glen Weyl 表示，此種作法可展現對民眾而言較重要的選項，能鼓勵民眾多了解不同項目，讓手持點數換算成選票達到最大效用，且避免將所有點數只投票給其中一項，造成票選結果失真。臺灣也在 2019 年總統盃黑客松的民眾票選活動首次引入平方投票法，便是期望能更真實的展現民意。

Elise 是某個民眾票選活動的其中一位選民，票選活動即是採用平方投票法。這個票選活動一共有 n 個提案，而每位選民同樣都提供 s 點。Elise 在瀏覽過所有提案以後，決定給第 i 個提案 **不超過** a_i 票。Elise 想要知道在使用最多 s 點，且在第 i 個提案投不超過 a_i 票的情形下，她最多能投幾票。

正式來說，假設最後 Elise 投給第 i 個提案 b_i 票，則 Elise 就是要求滿足以下兩個條件下 $b_1 + b_2 + \dots + b_n$ 的最大值：

- 對於所有介於 1 到 n 之間的正整數 i 都有 $b_i \leq a_i$ ，且 b_i 為非負整數。
- $b_1^2 + b_2^2 + \dots + b_n^2 \leq s$ 。

輸入格式

單筆輸入。第一行有兩個正整數 n, s ，代表提案數與每個人擁有的點數。

第二行有 n 個非負整數 a_1, a_2, \dots, a_n ， a_i 代表 Elise 在第 i 個提案最多想投幾票。

輸出格式

輸出一個整數，代表 Elise 最多能投幾票。

測資限制

- $1 \leq n \leq 3 \times 10^5$
- $1 \leq s \leq 10^{18}$
- $0 \leq a_i \leq 10^9$

輸入範例 1

5 18
1 4 0 3 2

輸出範例 1

8

輸入範例 2

1 10000000000000000000
999999999

輸出範例 2

999999999

輸入範例 3

5 100
1000000 1000000 1000000 1000000 1000000

輸出範例 3

22

備註

注意到 Elise 不需要把所有點數都用完。

在第一個範例中，Elise 可以投給五個提案分別 1, 2, 0, 3, 2 票，總共投了 8 票，需要 $1^2 + 2^2 + 0^2 + 3^2 + 2^2 = 18$ 點，剛好全部用完。

評分說明

本題共有 4 組測試題組，條件限制如下所示。每一組可有一或多筆測試資料，該組所有測試資料皆需答對才會獲得該組分數。

子任務	分數	額外輸入限制
1	12	$n = 1, 1 \leq s \leq 10^{12}, 0 \leq a_i \leq 10^6$ 。
2	33	$1 \leq s \leq 10^{12}$ ，且對於所有介於 1 到 n 之間的正整數 i 都有 $a_i = 10^6$ 。
3	22	$1 \leq s \leq 10^{12}, 0 \leq a_i \leq 10^6$ 。
4	33	無額外限制。

第六題：清除炸彈 (Bomb)

問題敘述

有一個 n 列 m 行的網格，行與列都從 1 開始編號，其中列由上到下編號，行由左至右編號。則每一格都可以用它的列編號與行編號兩個數字表示。網格中一共有 k 枚炸彈，每個格子最多只有一枚炸彈。

你想要清除盡量多的炸彈。你能做的唯一操作是，選定網格中的其中一個格子，並且將與這個格子同一行或同一列的格子裡的炸彈（如果有的話）全部清除。你只能做這個操作**恰好一次**。

以下面的範例（同時也是範例輸入 1）來說，炸彈分別位於 $(1, 2)$, $(2, 1)$, $(2, 5)$, $(3, 2)$, $(3, 4)$ 的格子中。如果你選了格子 $(1, 4)$ ，則只能清除位於 $(1, 2)$ 和 $(3, 4)$ 的炸彈；但若是選擇格子 $(2, 2)$ ，則能清除位於 $(1, 2)$, $(2, 1)$, $(2, 5)$, $(3, 2)$ 的 4 枚炸彈。






				
		(2,3)		
				(3,5)

Figure 1: 輸入範例 1 圖示，在此例中 $n = 3$, $m = 5$ 。炸彈圖源：Icon made by Freepik (www.freepik.com) from Flatcoin (www.flaticon.com)

請問你最多能清除幾枚炸彈？

輸入格式

單筆輸入。輸入第一行含有三個正整數 n, m, k ，前兩個整數代表網格大小為 n 列 m 行，第三個數字代表有幾枚炸彈。

接下來 k 行，每行有兩個正整數 r, c ，代表有一枚炸彈位於第 r 列第 c 行（即 (r, c) ）。炸彈的位置不會重複。

輸出格式

輸出一個整數，代表在選擇恰好一個格子情況下最多能清除幾枚炸彈。

測資限制

- $1 \leq n, m, k \leq 3 \times 10^5$
- $k \leq n \times m$
- $1 \leq r \leq n$
- $1 \leq c \leq m$

輸入範例 1

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

輸出範例 1

```
4
```

輸入範例 2

```
300000 300000 4
1 300000
300000 1
300000 300000
1 1
```

輸出範例 2

```
3
```

評分說明

本題共有 3 組測試題組，條件限制如下所示。每一組可有一或多筆測試資料，該組所有測試資料皆需答對才會獲得該組分數。

子任務	分數	額外輸入限制
1	27	$n, m \leq 300$ 。
2	35	$n, m \leq 3000$ 。
3	38	無額外限制。