

2019 網際網路程式設計全國大賽

國中組決賽

- 本次比賽共 7 題，含本封面共 16 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。

	題目名稱	時間限制 (秒)	記憶體限制
題目 A	貓咪會議	5	1024 MB
題目 B	拿行李	1	1024 MB
題目 C	字串匹配	1	1024 MB
題目 D	雜湊	1	1024 MB
題目 E	自動雞	1	1024 MB
題目 F	密碼生成器	1	1024 MB
題目 G	尋寶問題	3	1024 MB

2019 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. 貓咪會議

Problem ID: meeting

貓貓星球是個充滿貓貓的星球，星球上的每隻貓貓都十分的可愛，令人看了都忍不住想要對著貓貓們喵喵叫呢！

貓貓星球可以視為一個 2 維的 $X - Y$ 直角坐標平面，星球上一共有 N 隻貓貓，每隻貓貓都有他自己的家，分別位於 $(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)$ 。

俗話說的好，貓兒當自強！貓咪星球上的每隻貓貓每個月會有一次的聚會，殿王身為貓咪星球的主宰，必須為貓咪們決定聚會的地點。

因為每次聚會的重要性不盡相同，因此需要參與的貓貓數量也不一樣。

即將到來的聚會必須要有 M 隻貓貓參與，現在殿王想麻煩你，找出一個地點，使得離這個地點最近的前 M 隻貓咪的最長距離最小化。由於地點可能會有很多個，因此為了方便，殿王只要你幫忙算出這個距離即可。

對於兩隻位於 (x_1, y_1) 及 (x_2, y_2) 的貓咪，其距離為 $\sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$ 。

Input

輸入第一行有兩個整數 N, M ，代表貓貓星球上的貓貓數量，以及需要參與聚會的貓貓數量。

之後會有 N 行，每一行有兩個整數 x_i, y_i ，分別代表每隻貓貓位於貓貓星球中 (x_i, y_i) 的位置。

- $1 \leq M \leq N \leq 1000$
- $-10^3 \leq x_i, y_i \leq 10^3$

Output

輸出一個浮點數，表示殿王所求的最長距離最小可以多小。你輸出的答案與標準答案的相對或絕對誤差在 10^{-4} 以內即被判定為正確。

Sample Input 1

2 1 0 0 1 0	0.0
-------------------	-----

Sample Output 1**Sample Input 2**

2 2 0 0 1 0	0.5
-------------------	-----

Sample Output 2

B. 拿行李

Problem ID: luggage

喜歡出國玩的小 B 這次在等待被托運的行李時，突然想到了一個問題。

假設每個人有恰一件行李被托運，且行李輸送帶上最多只能有 k 件行李，且最多同時只能有 p 個人在輸送帶的周圍等待他的行李，且每個人都只會拿他自己的行李，且每個人只會且一定會在拿到自己的行李後離開輸送帶的周圍。現在，給定行李被放到輸送帶上的順序，以及每個人到達輸送帶周圍的順序，小 B 很好奇每個人能不能都順利的拿到自己的行李？

Input

輸入的第一行有三個非負整數 n, k, p ，意義如題目所述。輸入的第二行有 n 個正整數，且是 1 到 n 的一個排列，代表行李被放到輸送帶的順序。輸入的第三行也有 n 個正整數，且也是 1 到 n 的一個排列，代表每個人到輸送帶周圍開始等待自己的行李的順序。其中，第 i 個人在等待的是行李 i 。

- $1 \leq n, k, p \leq 10^5$

Output

輸出只有一行：如果每個人都能順利的拿到自己的行李，輸出 Yes，否則輸出 No。

Sample Input 1

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

Sample Output 1

```
Yes
```

Sample Input 2

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

Sample Output 2

```
No
```

This page is intentionally left blank.

C. 字串匹配

Problem ID: strings

這天，喜歡學習的小 B 學到了跟字串有關的演算法，也就是字串匹配。但是這個字串匹配的演算法跟常見的匹配有所不同，它回傳的訊息是兩個字串當中有幾個位置的字元是相同的，我們稱為「同位置字串匹配」。舉例來說，"NPSC" 跟 "NTU" 這兩個字串（不含雙引號）的同位置字串匹配是 1，因為兩者的相同之處只有第一個字元 N 是相同的。而 "abcdabc" 跟 "abcdcba"（不含雙引號）的同位置字串匹配是 5，因為這兩個字串在第 1, 2, 3, 4, 6 這五個位置的字元是一樣的。

現在，剛學完這個新潮字串演算法的小 B 興致勃勃地研究出了同位置字串匹配的進階版本：給定 n 個字串 s_1, s_2, \dots, s_n ，求出這 n 個字串當中兩兩字串的「同位置字串匹配」結果的加總。換句話說，假設兩個字串 a, b 的同位置字串匹配是 $f(a, b)$ ，那麼小 B 的演算法可以求出 $\sum_{i=1}^{n-1} \sum_{j=i+1}^n f(s_i, s_j)$ （也就是 $(f(s_1, s_2) + f(s_1, s_3) + \dots + f(s_1, s_n)) + (f(s_2, s_3) + f(s_2, s_4) + \dots + f(s_2, s_n)) + \dots + (f(s_{n-1}, s_n))$ ）是多少。

由於小 B 無法驗證他寫的這個新演算法的程式是否正確，於是小 B 前來尋求你的幫助。請幫助他算出每一筆問題的答案，讓小 B 知道他的程式是否正確。

Input

輸入的第一行是一個整數 n ，代表字串的數量。接下來的 n 行，每一行有一個字串 s_i 。

- $1 \leq n \leq 10^6$
- 每個字串的長度均介於 1 到 10^6 之間
- 每個字串均只包含大寫英文字母、小寫英文字母或是數字
- 所有字串的長度加總不會超過 5×10^6

Output

請輸出小 B 新研究出的演算法對於這 n 個字串作為輸入的輸出。

Sample Input 1

2 NTU NPSC	1
------------------	---

Sample Output 1**Sample Input 2**

5 meow Meow MEOw MeoW meoW	20
---	----

Sample Output 2

D. 雜湊

Problem ID: coder

小 B 是一個競賽選手。某天，他看到了一個雜湊函數。小 B 在看到那個雜湊函數的第一眼就深深的愛上了那個函數，原因當然是因為那個函數的變數居然使用了 meow！興奮的小 B 迫不及待的想要研究這個函數有沒有什麼有趣的性質。然而為了研究，小 B 需要知道很多字串在成為這個雜湊函數的參數之後，這個函數會回傳什麼樣的雜湊值。你，為了幫助小 B 的研究，決定偷偷的幫小 B 算出那些他想知道的雜湊值。

以下是那個小 B 一見鍾情的雜湊函數：

```
1 #include <stdint.h>
2 uint64_t MeowHash(char *s) {
3     uint64_t meow = 7122u1;
4     while (*s != 0)
5         meow = (meow << 13) ^ (meow >> 11) ^ (meow << 9)
6             ^ (meow >> 7) ^ (meow << 5) ^ (meow >> 3)
7             ^ (meow << 1) ^ ((uint64_t)(*s++) * 0xdeadbeef);
8     return meow;
9 }
```

Input

輸入只有一行，含有一個由大小寫英文字母所組成的字串 s ，即為小 B 想知道其雜湊值的字串。

- $1 \leq |s| \leq 1000000$

Output

輸出一行，包含一個數字，為輸入字串的雜湊值。

Sample Input 1

meow

Sample Output 1

7952703630401392505

Sample Input 2

Meow

Sample Output 2

1257661324266608215

E. 自動雞

Problem ID: chicken

貝斯比 (Bisbey) 是個養雞的農家。貝斯比一共使用四種廠牌的飼料來餵雞：N 牌、P 牌、S 牌與 C 牌。這四種廠牌的飼料各有各的優點，因此貝斯比會規劃雞接下來幾餐要吃哪一個廠牌的飼料。貝斯比規劃雞飼料的方法如下：貝斯比為每一隻雞都設置了一個長型的飼料槽，這個飼料槽的寬度恰好是一包飼料的寬度，最前面是雞吃飼料的地方，而最後面則是貝斯比把飼料放進飼料槽的地方。如此，貝斯比只要將雞接下來幾餐要吃的飼料一包一包依序從飼料槽最後面放進去，並且每次雞要吃飼料時從飼料槽的最前面拿出一包飼料給雞吃，就可以確保雞吃的飼料種類的順序是正確的了。

雖然貝斯比是養雞的，但是他最討厭的一件事就是定期要把雞飼料放進飼料槽中。為了解決這件事情，貝斯比想了各種方法，其中包含買一台自動餵雞的機器。然而，貝斯比最後採用了一個非常特別的方法：訓練雞自己把正確的飼料放進飼料槽。

經過長時間的訓練，貝斯比終於訓練出了第一批會自己把飼料放進飼料槽中的雞。具體來說，這些雞每吃完一包飼料，都會到飼料倉裡面把一或兩包飼料拿出來並從飼料槽後面放進去，如此飼料槽的飼料包數就永遠不會減少。由於這些雞自動自發，而且牠們又都很可愛，所以貝斯比決定把這些雞取名叫做「Q-自動雞」(Q-automa-chicken)。然而，貝斯比還沒有辦法訓練出可以規劃自己要吃甚麼的雞，所以他決定先觀察這些 Q-自動雞會把哪一種飼料拿出來吃。

經過觀察之後，貝斯比發現只要觀察出 Q-自動雞的「心情」，那麼 Q-自動雞拿出來的飼料種類是非常有規律的。具體來說，Q-自動雞總共有 M 種心情，並且在吃第一餐之前會是第 1 種心情。Q-自動雞每一餐都會從飼料槽的最前面拿一包飼料出來吃，吃完之後牠會根據當前的**心情和這一餐吃的飼料種類**，把一或兩包飼料從飼料槽後面放進去並改變自己的心情。另外，自動雞在沒有吃飼料的時候並不會改變自己的心情。

發現這個規律的貝斯比十分吃驚，因為他發現 Q-自動雞比他想像的還要聰明很多，而開始擔心 Q-自動雞會不會有某一天突然發動自動雞革命來推翻人類。因此，請你根據貝斯比觀察出的規則（也就是 Q-自動雞改變自己心情和放飼料的規律），寫出一個預測 Q-自動雞每天吃的飼料種類的程式，以證明人類創造的電腦至少沒有比雞還要差。

Input

第一行有兩個以空白隔開的正整數 N, M ，分別代表貝斯比想要知道的天數與 Q-自動雞總共有幾種心情。

第二行有一個只由 NPSC 四種字元組成的字串 S ，從左到右代表貝斯比一開始在飼料槽放的飼料種類（從最前面到最後面）。

接下來的 M 行描述貝斯比觀察到的規則。每行依序有四個正整數 x_N, x_P, x_S, x_C 和四個只由 NPSC 四種字元組成且長度為 1 或 2 的字串 y_N, y_P, y_S, y_C ，皆以空白隔開。對於其中的第 i 行，代表 Q-自動雞在第 i 種心情時吃下每種飼料之後的會如何改變心情和放哪種飼料： x_N, x_P, x_S, x_C 代表 Q-自動雞在吃下四種飼料（N 牌、P 牌、S 牌、C 牌）之後分別會變成哪一種心情； y_N, y_P, y_S, y_C 則代表 Q-自動雞在吃下四種飼料後分別會把哪種飼料放到飼料槽的最後面（如果有兩個字元，則先放第一個字元對應的飼料，再放第二個字元對應的飼料）。

- $N \leq 10000$
- $M \leq 100$
- S 的長度不超過 2000
- $1 \leq x_N, x_P, x_S, x_C \leq M$

Output

請輸出一個由 NPSC 四種字元組成、長度為 N 的字串於一行，其中第 i 個字元代表 Q-自動雞在第 i 餐吃的飼料種類。

Sample Input 1

```
16 1
N
1 1 1 1 P S C N
```

Sample Output 1

```
NPSCNPSCNPSCNPSC
```

Sample Input 2

```
16 2
NPSC
1 2 2 2 NN PP S CS
2 1 1 2 P C SS N
```

Sample Output 2

```
NPSCNNPPSSCSPPCP
```

F. 密碼生成器

Problem ID: password

小 Y 喜歡解題，為了解各式各樣的問題，他想要到各大 Online Judge 註冊帳號。

小 Y 很有資訊安全素養，他知道在很多地方用同一組密碼是非常不安全的。因此，他想產生 Q 組不同的密碼，使得每一組密碼都有不同的複雜度。不過，為了管理方便，小 Y 也希望生成的密碼長度皆為 N 。

一組密碼是一個只有 0 與 1 組成的字串。一組密碼的複雜度是「有至少一個 1 的子區間數量」。例如，密碼 010 的複雜度為 4，因為共有 $[1, 2], [1, 3], [2, 2], [2, 3]$ 這 4 個子區間有至少一個 1。

由於小 Y 正在認真解題，所以請你幫他撰寫一個密碼生成器，來生成指定複雜度的密碼。

Input

輸入的第一行有一個正整數 N, Q ，分別代表密碼的長度以及需要的密碼組數。接著 Q 行，第 i 行有一個非負整數 K_i ，代表第 i 組密碼的複雜度。

- $1 \leq N \leq 300$
- $1 \leq Q \leq 10^4$
- $0 \leq K_i \leq \frac{N \cdot (N+1)}{2}$

Output

輸出 Q 行，第 i 行有一組複雜度 K_i 的密碼。若不存在滿足條件的密碼，輸出一行 No。

Sample Input 1	Sample Output 1
4 10	No
1	No
2	No
3	0001
4	No
5	0100
6	1100
7	0101
8	1110
9	1111
10	

G. 尋寶問題

Problem ID: treasure

NPSC 國是一個得天獨厚的國家，國內盛產金礦以及銀礦，因此吸引了大量的遊客前往，大家都希望能在這裡挖到寶藏，一夜致富，這其中當然包含了厭倦了整天背單字的小 B。

具體來說，NPSC 國可以被表示成一個二維平面，且總共有 N 個金礦以及 M 個銀礦。金礦以及銀礦都有各自的位置以及價值，第 i 個寶藏位於 (x_i, y_i) 的位置，且價值為 v_i 。小 B 挖礦的方式十分特別：他會選擇兩個任意的寶藏（不管是金礦還是銀礦），並將在「以這兩個寶藏為對角線的矩形」內的所有寶藏都搜刮走。注意到，一旦選擇了對角線上的兩個寶藏，小 B 不能跳過任何一個在該矩形當中的寶藏！

當然小 B 也不能隨意挖礦。由於他最後還是得坐飛機把他挖到的寶藏帶回家，為了不超過飛機托運的重量限制，小 B 最多只能帶走 K 個金礦（銀礦的重量相較於金礦可以忽略）。小 B 就要出發了，請你幫忙估算一下他最多能挖到總價值為多少的寶藏。

Input

輸入第一行有三個整數 N, M, K ，分別代表金礦的數量、銀礦的數量，以及小 B 最多能擁有的金礦數量。

接著 $N + M$ 行，每行有三個正整數。其中第 i 行為 x_i, y_i, v_i ，代表第 i 個寶藏的位置以及價值。第 1 個到第 N 個為金礦，第 $N + 1$ 個到第 $N + M$ 個為銀礦。

- $1 \leq N + M \leq 5000$
- $0 \leq K \leq N$
- $1 \leq x_i, y_i, v_i \leq 10^9$
- 保證同一個位置至多只有一個寶藏。

Output

輸出一個整數代表小 B 最高能挖到總價值為多少的寶藏。

Sample Input 1

```
2 2 2
1 1 1
2 2 10
1 2 100
2 1 1
```

Sample Output 1

```
112
```

Sample Input 2

```
2 2 1
1 1 1
2 2 10
1 2 100
2 1 1
```

Sample Output 2

```
110
```

Sample Input 3

```
2 2 0
1 1 1
2 2 10
1 2 100
2 1 1
```

Sample Output 3

```
100
```