Question 0 – Hello World

(30 points)

Introduction

YTP Contest has started!

Let's verify everything first.
Is the internet setting correct?
Is the answer submit working well?
Did you use STDOUT to output your answers?

If everything is ready, 30 points are yours! Go! Go! Go!

Description

Please write a program to output Hello World!

Input Format

This problem requires no input.

Output Format

[A~Z][a~z], space, and common English punctuation.

Data Range

[A~Z][a~z], space, and exclamation mark "!".

Data Examples

Input Example 1
(no input)

Output Example 1
Hello World!

Example Explanation

Input Example 1 has no input, simply output Hello World!

送分題 – Hello World

(30 分)

前言
比賽開始了！

趕快驗證一下，
網路是否設定正確?
上傳競賽程式是否順利?
程式解答是否用 STDOUT 輸出？

都沒問題，30 分就到手了！ 繼續 … 衝！衝！衝！

問題描述
請寫一個程式輸出 Hello World!

輸入格式
本題無需輸入值

輸出格式
[A~Z][a~z], 空格, 以及及常用英文符號 。

資料範圍
[A~Z][a~z], 空格, 以及驚嘆號 “!”

資料範例
輸入範例 1
(無輸入值)

輸出範例 1
Hello World!

範例解釋
輸入範例 1, 無輸入值，簡單而快樂的輸出 Hello World!

Q1: Laptop Decoration

(5 points)

Description

To celebrate the success of YTP 2021, Mr. Y, Mr. T, and Mr. P decided to decorate their team's laptop with full YTP stickers. In order to be fully prepared for the contest, they came up with the following decoration design: They will choose an article and cut off all the characters from it. Then, assemble the characters into as many "YTP" as possible (case insensitive). Before they decorate their computer, they requested you, also participating in YTP 2021, to help them estimate how many "YTP" can they assemble from the chosen article.

Input Format

Read the input as the content of the article until EOF (end of file).

Output Format

Output a nonnegative integer represents the sets of "YTP" they can assemble from the article.

Data Range

The total number of characters in the article should be less than 1000.

Input Example 1

```
A YouTube Poop is a type of video mashup
or edit created by remixing/editing
pre-existing media sources often carrying
subcultural significance into a new video
for humorous, satirical, obscene and profane,
as well as annoying, confusing, scary,
shocking, surreal, or dramatic purposes.
```

Output Example 1

6

Example Explanation:

The number of case insensitive Y, T, P are 6, 11, 8 respectively. Thus, they can assemble at most 6 sets of "YTP".

問題 1 – 筆電裝飾 (Laptop Decoration)

(5 分)

問題敘述

為了慶祝 YTP 2021 能夠順利舉辦，小 Y 和小 T 和小 P 決定將他們隊伍的筆電貼上滿滿的 YTP。為了好好準備這次的比賽，他們想出了佈置筆電的方法，作法如下：他們找到了一篇英文文章，從文章中剪下所有的 Y、T、P 三個字母，用所有的字母拼出盡可能多的 YTP（不分大小寫）。他們在佈置電腦前，想要請同樣參加 YTP 的你，幫他們估計一下他們最多可以組合出幾組 YTP。

輸入格式

輸入文章內容，可能包含多個空白與換行，輸入到 EOF (檔案結尾) 為止。

輸出格式

輸出一個整數代表他們可以組合出的 YTP 數量

資料範圍

總字元數 $L \leq 1000$

輸入範例 1

```
A YouTube Poop is a type of video mashup
or edit created by remixing/editing
pre-existing media sources often carrying
subcultural significance into a new video
for humorous, satirical, obscene and profane,
as well as annoying, confusing, scary,
shocking, surreal, or dramatic purposes.
```

輸出範例 1

```
6
```

範例說明

在篇文章中不分大小寫的 Y、T、P 分別有 6、11、8 個，因此最多只能組合出 6 組 YTP。

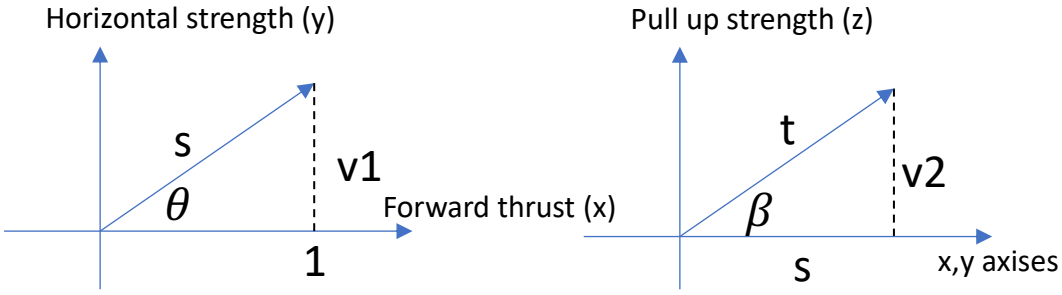
Q2: Unmanned Fighter

(5 points)

Description

To protect our airspace and save pilot's life, we need your help design an unmanned fighter which has strength in the region. Based on alliance's information from early this year, unmanned fighter can beat manned fighter easily in 9G circumstances. So, we need the program able to calculate spiral speed (t) and horizontal angle (theta) and pull up angle(beta) with given the unit thrust (as x-axis), for horizontal tilt (as y-axis), and pull up (z-axis) strength.

Related formula:



Related computation:

$\theta : theta$	$\beta : beta$
$\tan(\theta) = \frac{v1}{1} => \theta = \tan^{-1}\left(\frac{v1}{1}\right)$	$\tan(\beta) = \frac{v2}{s} => \beta = \tan^{-1}\left(\frac{v2}{s}\right)$
$\sin(\theta) = \frac{v1}{s} => v1 = s * sin(\theta)$	$\sin(\beta) = \frac{v2}{t} => v2 = t * sin(\beta)$
$\cos(\theta) = \frac{1}{s} => 1 = s * \cos(\theta)$	$\cos(\beta) = \frac{s}{t} => s = t * \cos(\beta)$
$s = \sqrt{v1^2 + 1^2} => v1 = \sqrt{s^2 - 1^2}$	$t = \sqrt{s^2 + v2^2} => v2 = \sqrt{t^2 - s^2}$
$\theta_{new} = \theta_{old} + \theta_{update}$	$\beta_{new} = \beta_{old} + \beta_{update}$

Input v1, v2, theta adjustment, beta adjustment, and output the (speed) t value and new v1 and new v2.

Input Format

v1,v2,theta adjustment, beta adjustment with comma separated.

Output Format

t,new_v1,new_v2
calculate the value and round to the second decimal place.

Data Range

0.0 <= v1, v2 <=10.00
0.01 <= theta adjustment, beta adjustment <= 0.10

1st Input Example

```
3.00,2.00,0.01,0.02
```

1st Output Example

```
3.74,2.97,2.06
```

1st Example Explanation:

v1=3.0, v2=2.0,

$\theta = \tan^{-1}(\frac{3.0}{1})$, s = $\sqrt{3^2 + 1^2} = \sqrt{10}$, $\beta = \tan^{-1}(\frac{2.0}{\sqrt{10}})$,

t = $\sqrt{2.0^2 + \sqrt{10}^2} = \sqrt{14}$

$\theta_{new} = \tan^{-1}\left(\frac{3.0}{1}\right) + 0.01,$

$\beta_{new} = \tan^{-1}\left(\frac{2.0}{t}\right) + 0.02$

$v2_{new} = \sqrt{14} * \cos(\beta_{new})$

$S_{new} = \sqrt{14} * \sin(\beta_{new})$

$v1_{new} = S_{new} * \cos(\theta_{new})$

2nd Input Example

```
3.10,2.10,0.02,0.01
```

2nd Output Example

```
3.88,3.10,2.13
```

3rd Input Example

```
2.10,2.10,0.02,0.02
```

3rd Output Example

```
3.13,2.08,2.15
```

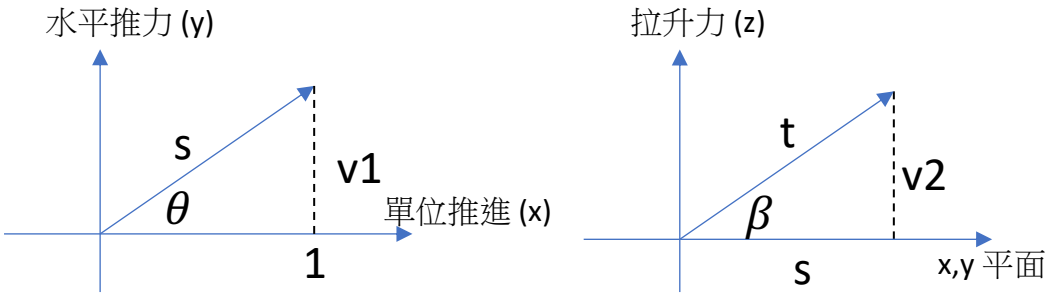
問題 2 – 無人戰機 (Unmanned Fighter)

(5 分)

問題敍述

為了保障一直被侵擾的領空並避免戰機飛行員的折損，需要委託你協助設計具有區域競爭力的無人戰機，依據今年友邦的經驗無人戰機在 9G 離心力的纏鬥下能輕易戰勝人類駕駛，因此需要你寫一個程式計算螺旋飛行 (Spiral) 在單位推進 (x) 下平移 (y) 和拉升 (z) 的可讓戰機往前進的速度 (t) 及水平傾斜的角度 (theta) 和上升傾斜的角度 (beta)。

主要公式與原理如下:



相關數學計算:

$\theta : theta$	$\beta : beta$
$\tan(\theta) = \frac{v1}{1} => \theta = \tan^{-1}\left(\frac{v1}{1}\right)$	$\tan(\beta) = \frac{v2}{s} => \beta = \tan^{-1}\left(\frac{v2}{s}\right)$
$\sin(\theta) = \frac{v1}{s} => v1 = s * sin(\theta)$	$\sin(\beta) = \frac{v2}{t} => v2 = t * sin(\beta)$
$\cos(\theta) = \frac{1}{s} => 1 = s * \cos(\theta)$	$\cos(\beta) = \frac{s}{t} => s = t * \cos(\beta)$
$s = \sqrt{v1^2 + 1^2} => v1 = \sqrt{s^2 - 1^2}$	$t = \sqrt{s^2 + v2^2} => v2 = \sqrt{t^2 - s^2}$
$\theta_{new} = \theta_{old} + \theta_{update}$	$\beta_{new} = \beta_{old} + \beta_{update}$

輸入 v1, v2, theta 修正值, beta 修正值。要回傳 t 值和新的 v1, v2 值。

輸入格式

v1,v2,theta 修正值, beta 修正值，中間以逗號隔開。

輸出格式

輸出 t, new_v1,new_v2 的值，四捨五入到小數第二位。

資料範圍

0.00 <= v1, v2 <=10.00,
0.01 <= theta adjustment, beta adjustment <= 0.10

輸入範例 1

```
3.00,2.00,0.01,0.02
```

輸出範例 1

```
3.74,2.97,2.06
```

範例 1 說明

v1=3.0, v2=2.0,

$\theta = \tan^{-1}(\frac{3.0}{1})$, $s = \sqrt{3^2 + 1^2} = \sqrt{10}$, $\beta = \tan^{-1}(\frac{2.0}{\sqrt{10}})$,

$t = \sqrt{2.0^2 + \sqrt{10}^2} = \sqrt{14}$

$\theta_{new} = \tan^{-1}\left(\frac{3.0}{1}\right) + 0.01,$

$\beta_{new} = \tan^{-1}\left(\frac{2.0}{t}\right) + 0.02$

$v2_{new} = \sqrt{14} * \cos(\beta_{new})$

$S_{new} = \sqrt{14} * \sin(\beta_{new})$

$v1_{new} = S_{new} * \cos(\theta_{new})$

輸入範例 2

```
3.10,2.10,0.02,0.01
```

輸出範例 2

```
3.88,3.10,2.13
```

輸入範例 3

```
2.10,2.10,0.02,0.02
```

輸出範例 3

```
3.13,2.08,2.15
```

Q3: Hungry Time

(10 points)

Description

Little B not only enjoys eating ramen but also has a strict standard about it. A few days before, YTP Ramen opened and attracted Little B's attention. He have eaten N bowls of ramen there. If we write them as a sequence according to the time Little B ate it, we get a sequence of length N, called a_i , for all i between 1 and N. To score YTP Ramen, he wants to compute the "delicious rate" of it. The "delicious rate" is defined as the following:

- We defined a sequence of length $N-1$ which satisfies $b_i = a_{i+1} - a_i$, forall i between 1 and $N-1$.
- The "delicious rate" of YTP Ramen is $b_1 \times b_2 \times b_3 \times \ldots \times b_{N-1}$

Little B is busy eating ramen. Can you help him?

Otherwise, the answer may be very large. For convenience, if the answer is X, please output Ysuch that $0 \leq Y < 10^9 + 9, X - Y = (10^9 + 9) \times k$, where k is an integer. It can be proved that only one integer Ysatisfies these conditions.

Input Format

The first line of input consists of one integer N, denoting the number of ramen he ate at YTP Ramen.

The next line consists of N integers, denoting the score of each ramen he ate.

Output Format

Output a nonnegative integer represents the "delicious rate".

Data Range

- $2 \leq N \leq 10^6$
- $1 \leq a_i \leq 10^9$, for all i between 1 and N

Input Example 1

```
3
1 1 1
```

Output Example 1

```
0
```

Input Example 2

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

Output Example 2

1

Input Example 3

4
1 2 4 1

Output Example 3

1000000003

Example Explanation:

In Example 1, the sequence $b = \{0,0\}$. The "delicious rate" is $0 \times 0 = 0$.

In Example 2, the sequence $b = \{1,1,1,1\}$. The "delicious rate" is $1 \times 1 \times 1 \times 1 = 1$.

In Example 3, the sequence $b = \{1,2,-3\}$. The "delicious rate" is $1 \times 2 \times -3 = -6$, but -6 does not satisfy $0 \leq Y < 10^9 + 9$. Instead, the answer is $10^9 + 3$. We can easily know $10^9 + 3$satisfy all the conditions.

問題 3 – 飢餓時刻 (Hungry Time)

(10 分)

問題敘述

小 B 是個拉麵愛好者，同時也有嚴格的拉麵評分標準。幾天之前，YTP 拉麵開張了，這讓小 B 很感興趣。小 B 已經去店裡吃了 N 碗拉麵。如果我們根據小 B 吃拉麵的時間，把他對每一碗拉麵的評分寫下來，可以得到一個長度為 N 的序列 a_i，對於所有介在 1 到 N之間的 i。 為了給 YTP 拉麵一個全面的評價，小 B 想要計算這家店的「美味度」。「美味度」是這樣定義的：

- 我們定義一個長度為 $N-1$ 的序列 b，滿足 $b_i = a_{i+1} - a_i$，對於所有介在 1 到 $N-1$之間的 i
- YTP 拉麵的「美味度」就是 $b_1 \times b_2 \times b_3 \times \ldots \times b_{N-1}$

小 B 正忙著吃拉麵，你能幫他計算 YTP 拉麵的「美味度」嗎？

除此之外，答案可能會非常的大。為了方便起見，如果答案是 X，請輸出 Y滿足 $0 \leq Y < 10^9 + 9$, 且 $X - Y = (10^9 + 9) \times k$，此處 k 是一個整數。可以證明滿足這些條件的 Y只有一個。

輸入格式

輸入的第一行包含一個正整數 N，表示小 B 吃過幾碗 YTP 拉麵。

下一行包含 N 個正整數（以空格分隔），代表每碗拉麵的評分。

輸出格式

輸出一個非負的整數代表 YTP 拉麵的「美味度」。

資料範圍

- $2 \leq N \leq 10^6$
- $1 \leq a_i \leq 10^9$，對於所有介在 1 到 N之間的 i

輸入範例 1

```
3
1 1 1
```

輸出範例 1

```
0
```

輸入範例 2
5
1 2 3 4 5

輸出範例 2
1

輸入範例 3
4
1 2 4 1

輸出範例 3
1000000003

範例說明
在範例 1，序列 $b = \{0,0\}$。故「美味度」是 $0 \times 0 = 0$.

在範例 2，序列 $b = \{1,1,1,1\}$。故「美味度」是 $1 \times 1 \times 1 \times 1 = 1$.

在範例 3，序列 $b = \{1,2,-3\}$。故「美味度」是 $1 \times 2 \times -3 = -6$，但 -6不滿足 $0 \leq Y < 10^9 + 9$。作為替代，答案是 $10^9 + 3$。我們可以輕易地知道 $10^9 + 3$滿足所有的條件。

Q4: Chibi

(10 points)

Description

During the "Three Kingdoms" era, the "Battle of Chibi" is one of the most famous battles. Shu Han and Eastern Wu won this battle with the help of their best military advisers.

During this war, "chained ships" is one of the most classic strategies. Cao Cao chained his ships from stem to stern. When Huang Gai drove his fire ships to those chained ships, all Cao Cao's ships got fire sequentially.

After reading the story above, Little Y feels very interesting. He wants to write a program to simulate the above process.

Cao Cao's chained ships can be seen as integer points on a one-dimension number line. To simplify the problem, we number Cao Cao's boats as 1, 2, 3, ……, M. Little Y will play Huang Gai's role, sending N fire ships to those chained ships. The i-th fire ship arrives at position x_i on the t_i-th time. The chained ship is on fire immediately. The fire will spread to both directions at a speed of one boat per unit time.

Now, Little Y is curious that what is the time that each chained ship is on fire.

Input Format

The first line of the input contains two integers N, M, denoting the number of fire ships Little Y sent, and the number of Cao Cao's chained ships.

In the next M lines, the i-th line contains two integers x_i, t_i, denoting that the i-th fire ship will arrive on position x_i in time t_i.

Output Format

Print M space-separated integers in one line. The i-th integer denotes the time that the chained ship on position i is on fire.

Data Range

- $1 \leq N \leq 100$
- $1 \leq M \leq 100$
- $1 \leq x_i \leq M$
- $1 \leq t_i \leq 100$

Input Example 1

```
1 6
2 1
```

Output Example 1

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

Input Example 2

```
2 6
2 1
5 1
```

Output Example 2

```
2 1 2 2 1 2
```

Input Example 3

```
2 6
2 1
5 10
```

Output Example 3

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

Example Explanation:

In example 1, the fire ship arrives at position 2 on time 1. The fire spreads in both directions at a speed of 1 unit length / 1 unit time. So the answer is 2 1 2 3 4 5.

In example 2, on time 1, the ships on position 2, 5 is on fire. The fire spreads in both directions simultaneously. So on time 2, all ships are on fire.

In example 3, the arrival time on fire ship 2 is too late, so all the chained ships are on fire of fire ship 1.

問題 4 – 赤壁之戰 (Chibi)

(10 分)

問題敍述

在三國時代中，赤壁之戰是一場非常著名的戰役。在吳、蜀兩國軍師的運籌帷幄之下，成功的拿下這場經典的戰役。

而在這場戰役中，「連環計」堪稱經典之作。曹操軍把在水上的船通通串在一起，而黃蓋開著載著火藥的船，把火藥往其中一艘船丟，所有的船就依序燒了起來。

看到以上的故事後，小 Y 覺得非常的有趣，於是他想要寫一個程式，來模擬這個過程。

曹操的船，可以視為一維數線上的連續整數點。為了方便起見，我們把這些點編號成 1, 2, 3, ……, M 。而小 Y 總共會派送 N 艘載著火藥的船去攻擊，第 i 艘船會在時間 t_i 抵達數線上的 x_i 位置，並且在抵達的當下，會立刻點燃該位置的船。船上的火會以每秒一艘船的速度，向左右擴散。

現在，小 Y 很好奇，每艘船最早在什麼時間點，會被火藥點燃。

輸入格式

輸入的第一行包含兩個正整數 N, M ，分別代表小 Y 派出載著火藥的船的數量，以及曹操串在一起的船的數量。

接下來的 N 行，第 i 行會包含兩個整數 x_i, t_i ，代表小 Y 派出的第 i 艘船，會在時間 t_i 抵達數線上 x_i 的位置。

輸出格式

輸出 M 個用空白隔開的整數於一行，第 i 個整數代表位於位置 i 的船著火的時間。

資料範圍

- $1 \leq N \leq 100$
- $1 \leq M \leq 100$
- $1 \leq x_i \leq M$
- $1 \leq t_i \leq 100$

輸入範例 1

```
1 6
2 1
```

輸出範例 1

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

輸入範例 2

```
2 6
2 1
5 1
```

輸出範例 2

```
2 1 2 2 1 2
```

輸入範例 3

```
2 6
2 1
5 10
```

輸出範例 3

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

範例說明

在範例一中，第一艘船在時間點 1 的時候抵達連環計中的位置 2 ，火以 1 距離 / 1 時間單位的速度向左右擴散，因此每艘船被火燒到的時間為 2 1 2 3 4 5。

在第二個範例中，時間 1 的時候，位於 2, 5 的船就已經著火，火會朝著兩個方向傳遞，因此在時間 2 的時候，剩下的船就通通著火了。

在第三個範例中，由於第二艘船抵達的時間過晚，因此第一艘船就已經把所有的船通通著火了。

Q5: Squares

(15 points)

Description

In a Cartesian coordinate system, there are nm lattice points (x, y) (points such that both coordinates are integers) satisfy $1 \leq x \leq n$ and $1 \leq y \leq m$. Consider squares that have their four corners on these $n \times m$ points. What is the sum of the areas of these squares?

Input Format

The input consists of one line, containing 2 integers n and m.

Output Format

Output a single non-negative integer on a single line, which is the area sum of the squares.

Data Range

$$1 \leq n, m \leq 1000$$

Input Example 1

3 3

Output Example 1

10

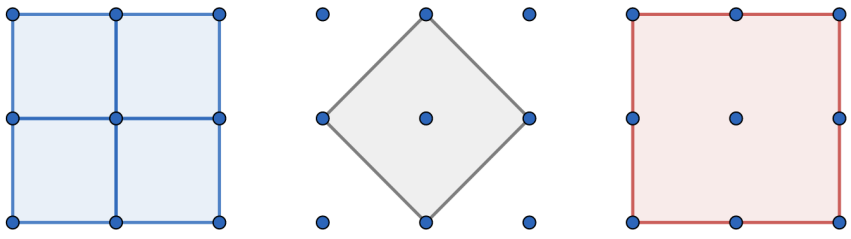
Input Example 2

1000 1000

Output Example 2

11111138888850000

Example Explanation:



For example 1, as the above figure shows, these 9 points can form four 1×1 squares, a 2×2square, and a $\sqrt{2} \times \sqrt{2}$ square which is tilted 45 degrees. The sum of areas is $4 \times 1 + 1 \times 4 + 1 \times 2 = 10$.

問題 5 – 正方形 (Squares)

(15 分)

問題敘述

在直角座標系中，所有滿足 $1 \le x \le n$ 和 $1 \le y \le m$ 的格點 (x, y) 會形成一個 $n \times m$的格點。我們可以從這 $n \times m$ 個點中挑出四個相異的點形成一個正方形，請求出所有可能的正方形的面積總和。

輸入格式

輸入只有一行，包含兩個正整數 n, m ，以空白間隔，代表格點的長寬。

輸出格式

輸出一個非負整數，代表所有正方形的面積總和。

資料範圍

$1 \le n, m \le 1000$

輸入範例 1

3 3

輸出範例 1

10

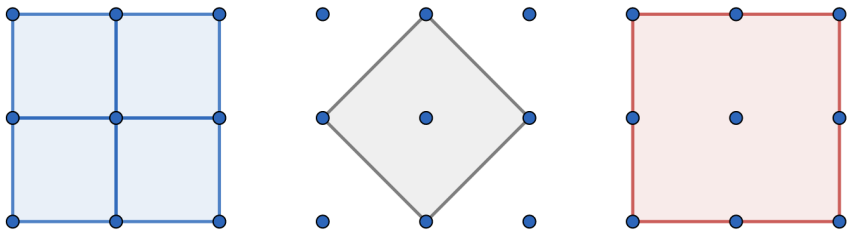
輸入範例 2

1000 1000

輸出範例 2

11111138888850000

範例說明



如上圖，3×3 的格點可以形成 4 個 1×1 的正方形、一個 2×2的正方形，和一個傾斜 45 度的 $\sqrt{2} \times \sqrt{2}$正方形，面積總和為 $4 \times 1 + 1 \times 4 + 1 \times 2 = 10$。

Q6: Dig In

(15 points)

Description

After the Salmon Sushi Event, Sushi Chang held a sushi eating event. The sushi store launched some evil discounts. They put some sushi which people do not enjoy eating together on a plate. If you manage to eat all of the sushi on the plate at once, you'll receive a discount. Of course, as a picky customer, you can order only part of the set (and pay just the partial price) without ordering the whole set of sushi if you like.

Today, Sushi Chang prepared N sushies and M plates. The N sushies will have its own price and weight. There are S_i sushies and a X_i yen discount ticket on the i th plate. If you eat all the sushi on the plate, you will get a X_i yen discount at checkout. As a super-heavyweight sushi eater, you are born to eat as much sushi by weight as you can. What is the maximum weight of the sushi you can eat if your budget is C yen?

Note that all the N sushies can only be ordered once (whether à la carte or within a set), which means if you order one in a set, you can't order the same one anymore in any other setting. Also you can just order some of the sushi on any plate, even one that isn't included in any set. Please note that discounts will only be given if you order everything in the set.

Input Format

The first line contains three integers N, M, C representing the number of sushies, sets, and budget
The second line contains N integers, $P_1 \ldots P_N$, P_i ,Pi represents the price of the ith sushi.
The third line contains N integers, $W_1 \ldots W_N$, W_i represents the weight of the i th sushi.
The fourth line contains M integers $X \ldots X_N$, $X_i$represents the discount for the ith set.
The fifth line contains M integers$S_1 \ldots S_N$, S_i represents the number of slushies on the ith set.
For the following M lines, the i th line contains S_i numbers $I_{i1} \ldots I_{iS_i}$ represents the indices of slushies on the i th plate.

Output Format

Output an integer, which is the weight you can eat within the budget.

Data Range

- $1 \leq M \leq N \leq 1000$
- $1 \leq P_i, C \leq 100000$
- $1 \leq W_i \leq 1000000$
- $1 \leq X_i \leq 100000$
- $1 \leq S_i \leq N$

It is guaranteed that all I_{ij} is unique and $1 \leq I_{ij} \leq N$. Also, the discount will not be greater than the total price on the plate.

Note that, it's not guaranteed that every sushi appears in a set. (Though you can still eat them if they're not in any set)

Input Example 1

```
5 2 5
2 3 4 1 10
4 1 2 9 2
5 2
3 2
1 2 3
4 5
```

Output Example 1

```
16
```

Example Explanation:

For this example, there are two sets. We can choose the first set and the fourth sushi. It costs $2 + 3 + 4 + 1 - 5 = 5$ yen, and you will receive $4 + 1 + 2 + 9 = 16$ weight.

Input Example 2

```
5 2 100
25 12 34 41 40
43 74 90 37 2
9 16
2 2
4 1
3 5
```

Output Example 2

```
209
```

問題 6 – 吃爆髒壽司 (Dig In)

(15 分)

問題敘述

在鮭魚風暴後，髒壽司店有舉辦了一個吃壽司的活動。這家壽司店推出了一些邪惡優惠組合，把一些同時吃下去會感到不舒服的壽司，放在同個盤子上一起賣。只要你能一起吃下盤子上的所有壽司，就會給你一個折扣。當然，作為一個良心店家，折扣絕對可以當次抵用。當然，作為一個奧客的你，當然可以只單點盤子上的一部分壽司，但是不享有優惠。

今天髒壽司準備了 N 貫壽司和 M個放在盤子上的套餐，這 N 貫壽司會有各自的價錢和重量。第 i 個盤子上會放有 S_i 貫壽司，和一張 X_i円的折價卷，如果一次吃了盤子上的所有壽司，結帳時價格就會減掉X_i円。身為一個重量級吃壽司選手的你，在預算內吃最大重量是你的使命，請問 C円最多可以吃到多少重量的壽司？

注意這邊 N 貫壽司都是獨一無二的，也就是如果你在套餐點了它，就不能再單點同樣壽司，反過來也是一樣。並且，你可以單點任何被放在盤子上的壽司，或是沒有被放在任何盤子上的壽司。

輸入格式

第一行有三個整數 N, M, C分別代表壽司數量以及套餐數量以及預算
第二行有 個整數 $P_1 \ldots P_N$ ，P_i代表第 i 貫壽司的價格
第三行有 N 個整數 $W_1 \ldots W_N$ ，W_i 代表第 i 貫壽司的重量
第四行有 N 個整數 $X \ldots X_N$ ，X_i 代表第 i 個套餐的優惠卷價值
第五行有 M 個整數 $S_1 \ldots S_N$ ，S_i 代表第 i 個套餐的壽司數量
接下來 M 行其中第 i 行有 S_i 個整數 I_{ij}代表第 i 盤子裝有第 I_{ij} 個壽司

輸出格式

輸出一個整數 W 代表預算下使用優惠卷可以吃下的最大重量

資料範圍

- $1 \leq M \leq N \leq 1000$
- $1 \leq P_i, C \leq 100000$
- $1 \leq W_i \leq 1000000$
- $1 \leq X_i \leq 100000$
- $1 \leq S_i \leq N$

保證 l_{ij} 不重複，$1 \leq l_{ij} \leq N$，且優惠券價值不會比盤子上的壽司總價值便宜。
注意： 不保證所有壽司都有被放在盤子上（雖然身為奧客也可以點不在盤子上的壽司）

輸入範例 1

```
5 2 5
2 3 4 1 10
4 1 2 9 2
5 2
3 2
1 2 3
4 5
```

輸出範例 1

```
16
```

範例說明

這個例子一共有兩種套餐，我們可以選擇吃下第一個套餐的三個壽司，和單獨吃第四個壽司。這樣我們總共需要花 $2+3+4+1-5=5$ 円，然會得到 $4+1+2+9=16$ 重量。

輸入範例 2

```
5 2 100
25 12 34 41 40
43 74 90 37 2
9 16
2 2
4 1
3 5
```

輸出範例 2

```
209
```

Q7: Peak

(20 points)

Description

Given an array $a_1, a_2, \ldots, a_n$ of length n. (i, j, k) is called a peak if $i < j < k$, $a_j > a_i$, and $a_j > a_k$. Please count the number of peak (i, j, k)such that $a_i + a_j + a_k$ is an even number.

Input Format

The input consists of two lines. The first line contains a single integer n, and the second line contains the array $a_1, a_2, \ldots, a_n$, with the numbers separated with a single space.

Output Format

Output an integer, which is the number of peaks that satisfy the requirement.

Data Range

- $1 \leq n \leq 200000$
- $1 \leq a_i \leq 10^9$
- if $i \neq j$, then $a_i \neq a_j$

Input Example 1

```
4
1 3 4 2
```

Output Example 1

```
1
```

Input Example 2

```
5
10 20 30 40 50
```

Output Example 2

```
0
```

Input Example 3

```
5
1 2 100 3 4
```

Output Example 3

```
2
```

Example Explanation:

For example 1, there are 3 peaks – $(1, 2, 4)$, $(1, 3, 4)$, $(2, 3, 4)$ (the numbers here are the indices, not the element of the array), and the only peak that has an even sum is $(1, 2, 4)$, which corresponds to the numbers $1, 3, 2$, as $1 + 3 + 2$ is an even number.
For example 2, there are no peaks in the array.
For example 3, all peaks involves the middle element, 100, and we have to choose an element on the left and an element on the right that sums to an even number. There are two possible ways.

問題 6 – 尖峰 (Peak)

(20 分)

問題敘述

給定長度為 n 的相異元素陣列 $a_1, a_2, \ldots, a_n$。若數對 (i, j, k) 滿足$i < j < k$ 且 $a_j > a_i$, $a_j > a_k$，則我們稱此數對是一個尖峰。請計算滿足 $a_i + a_j + a_k$ 是偶數的尖峰 (i, j, k) 數量。

輸入格式

輸入有兩行。第一行有一個正整數 n，第二行有 n 個用空白分隔的正整數 $a_1, a_2, \ldots, a_n$。

輸出格式

輸出一個整數代表滿足條件的尖峰數量。

資料範圍

- $1 \leq n \leq 200000$
- $1 \leq a_i \leq 10^9$
- 若 $i \neq j$ 則 $a_i \neq a_j$

Input Example 1

```
4
1 3 4 2
```

Output Example 1

```
1
```

Input Example 2

```
5
10 20 30 40 50
```

Output Example 2

```
0
```

Input Example 3

```
5
1 2 100 3 4
```

Output Example 3

```
2
```

範例說明

範例一，總共有三個尖峰：$(1, 2, 4)$, $(1, 3, 4)$, $(2, 3, 4)$（這些數字是陣列元素位置，並不是陣列中的元素值），唯一滿足條件的尖峰位置是 $(1, 2, 4)$，對應到元素值是 $1, 3, 2$，因為 $1 + 3 + 2$是偶數，所以輸出 1。

範例二，沒有任何的尖峰。

範例三，總共有四個尖峰：$(1, 3, 4)$, $(1, 3, 5)$, $(2, 3, 4)$,$(2, 3, 5)$，元素值的和是偶數的尖峰位置有 $(1, 3, 4)$，$(2, 3, 5)$ ，所以輸出 2。

Q8: No Plagiarism!

(20 points)

Description

Prof. Aqua is the Instructor of the course “Introduction To Brainfuck”. If you haven’t heard about Brainfuck, it is an esoteric programming language created in 1993 by Urban Müller, and it only consists of eight commands, listed below.

Character	Meaning
>	Increment the data pointer (to point to the next cell to the right).
<	Decrement the data pointer (to point to the next cell to the left).
+	Increment (increase by one) the byte at the data pointer.
-	Decrement (decrease by one) the byte at the data pointer.
.	Output the byte at the data pointer.
,	Accept one byte of input, storing its value in the byte at the data pointer.
[	If the byte at the data pointer is zero, then instead of moving the instruction pointer forward to the next command, jump it *forward* to the command after the *matching*] command.
]	If the byte at the data pointer is nonzero, then instead of moving the instruction pointer forward to the next command, jump it *back* to the command after the *matching* [command.

Since the language is very hard to read and write, every time when there is an assignment, most of the students “borrowed” other students’ code “for reference.” Of course, Prof. Aqua is not happy about this situation, so she asked you to help her detect plagiarism.
Fortunately, you don’t need to learn Brainfuck to do this. Now given all students’ code submissions, and then Prof. Aqua will ask you the similarity of student i’s code and student j’s code. To simplify the problem, you only need to find the length of the longest common prefix of their codes.

Input Format

In the first line consists of two positive integers $n$and m, separated by space.
In the next $n$lines, each line is a string only consisting of the 8 Brainfuck commands, which represents the code submission of student 0, 1, …, $n-1$, respectively.
The next $m$lines, each line consists of two integers i, j ranging from 0 to $n-1$(inclusive), which represents that Prof. Aqua asks you the length of the longest common prefix of student i's and student j's code submissions.

Output Format

For each query (i, j) Prof. Aqua asked, output the length of the longest common prefix of student i and student j's code submissions.

Data Range

$1 \leq n, m \leq 1000000$
The sum of length of all code submissions is no more than 1000000.

Input Example 1

```
2 1
++--><[]
++-++<>[]
0 1
```

Output Example 1

```
3
```

Input Example 2

```
3 3
+++++
+++++--
+++++-
0 1
0 2
1 2
```

Output Example 2

```
5
5
6
```

Input Example 3

```
4 5
><><><><
++++++++
------++
++++++--
0 1
0 3
1 2
1 3
2 3
```

Output Example 3

```
0
0
0
6
0
```

Example Explanation:

For example 1, the longest common prefix of “++--><[]” and “++-++<>[]” is “++-”, so it outputs 3.
For example 2, code 0 and 1, code 0 and 2 share the prefix “+++++”, so the first two lines of output are both 5. Code 1 and 2 share the prefix “+++++-”, so the third line of output is 6.
For example 3, queries 1, 2, 3, 5 all only have the common prefix “” which is an empty string, so the corresponding lines of output are all 0. Query 4 has the longest common prefix “++++++”, so the fourth line of output is 6.

問題 8 – 不能抄襲！(No Plagiarism!)

(20 分)

問題敘述

阿夸教授是「Brainfuck 導論」的任課教師。如果你沒聽過 Brainfuck 的話，Brainfuck 是一種極小化的程式語言，它是由 Urban Müller 在 1993 年創造的。這種語言由八種運算子構成，列表如下：

字元	含義
>	指標加一
<	指標減一
+	指標指向的位元組的值加一
-	指標指向的位元組的值減一
.	輸出指標指向的單元內容（ASCII 碼）
,	輸入內容到指標指向的單元（ASCII 碼）
[	如果指標指向的單元值為零，向後跳轉到對應的] 指令的次一指令處
]	如果指標指向的單元值不為零，向前跳轉到對應的 [指令的次一指令處

因為這個語言實在是太難了，每次要繳交作業時，大多數學生都會向其他同學「借」作業來「參考」。阿夸教授當然不樂見這種狀況，因此她拜託你來幫她抓抄襲。
幸運的是，你不需要學會 Brainfuck 也能達成這項任務。給你所有學生繳交的程式碼，然後阿夸教授會問你學生i和j的程式碼相似程度。為了簡化問題，你只需要找出他們的程式碼的最長共同前綴就好。

輸入格式

第一行有兩個整數 n和 m，以空白分隔。
接下來 n行，每行都是一個由 Brainfuck 指令組成的字串，代表學生 0, 1, …, $n-1$提交的程式碼。
接下來 m行，每行有兩個整數 i, j介於 0 到 $n-1$之間，代表阿夸教授詢問你學生 i和 j的程式碼相似程度。

輸出格式

對每個詢問輸出一個整數後換行，代表詢問的最長共同前綴。

資料範圍

$1 \leq n, m \leq 1000000$
所有程式碼的長度總和不超過 1000000

輸入範例 1

```
2 1
++--><[ ]
++-++<>[ ]
0 1
```

輸出範例 1

```
3
```

輸入範例 2

```
3 3
+++++
+++++--
+++++-
0 1
0 2
1 2
```

輸出範例 2

```
5
5
6
```

輸入範例 3

```
4 5
><><><><
++++++++
------++
++++++--
0 1
0 3
1 2
1 3
2 3
```

輸出範例 3

```
0
0
0
6
0
```

範例說明

第一個範例的“++--><[ ]” 和 “++-++<>[ ]” 的最長共同前綴是 “++-”，長度為 3，因此輸出 3。

第二個範例的程式碼 0, 1 和程式碼 0, 2 皆有前綴 “+++++”，所以前兩行輸出皆為 5。程式碼 1, 2 皆有前綴 “+++++-”，所以第三行輸出是 6.

對於第三個範例，詢問 1, 2, 3, 5 的最長共同前綴都是空字串，所以對應到的輸出皆為 0。

第四個詢問的最長共同前綴是”++++++”，所以第四行輸出是 6。

Q9: Messy Toys

(25 points)

Description

Zisk, a child prodigy, who is very passionate about Computer Science. Therefore, he requested to build his home in the shape of a tree of Computer Science with N nodes. Generally speaking, there are exactly $N-1$ corridors connect the N rooms of his home. Each of the corridors bidirectionally connects two rooms, which satisfies that each pair of the two rooms can pass through a unique series of corridors to go back and forth.

However, the child prodigy also needs to relax and play games. Today, Zisk found his toys messed up when he was playing a game. Originally, each of these toys has its own unique room it belongs to. Also, each of the rooms has exactly a toy in there.

Zisk wants to put these toys back in place one by one. **And then return to the room where he is in at the beginning to continue playing his game**. Merely, he is too lazy to make moves. Hence, he thought of a question before tidying up these toys, as a fanatic of Computer Science. What is the minimum number of **moves** he can make if he only does the following four operations?

The four operations are:

1. Pick up the toy in the room he is in when he has no toy in his hand.
2. Put the toy in the room he is in when he has a toy in his hand and the room he is in has no toy in it.
3. Swap the toy in the room he is in with the toy in his hand when he has a toy in his hand and the room he is in has a toy in it.
4. Move to another room by passing through a corridor directly connect to the room he is in.

In other words, Zisk wants to minimize the number of the 4th operations to put all the toys back in place. And then return to the room where he is in at the beginning to continue playing his game.

Zisk knows that you love Computer Science, too. So, he decides to duel with you, for seeing who can solve this problem earlier.

Input Format

The first line of the input contains two integers N, X, indicating that Zisk's home is a shape of a tree with N nodes, and Zisk and his gaming console are at room X.

The next line contains N integers $p_1, \ldots\ldots, p_N$ separated by spaces, indicating that the toy in room i should be put back into room p_i.

Next, followed by $N-1$ lines, each line contains two integers u_i, v_i, indicating that there is a corridor connects room u_i and room v_i.

Output Format

Output an integer, indicating the minimum number of moves can Zisk make to put all the toys back in place. And then return to the room where he is in at the beginning to continue playing his game.

Data Range

- $1 \leq N \leq 600$
- $1 \leq X, u_i, v_i \leq N$
- $p_1 \sim p_N$ is a permutation of $1 \sim N$
- It is guaranteed that the shape of Zisk's home is a valid tree with N nodes

Input Example 1

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

Output Example 1

```
6
```

Input Example 2

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

Output Example 2

```
12
```

Input Example 3

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

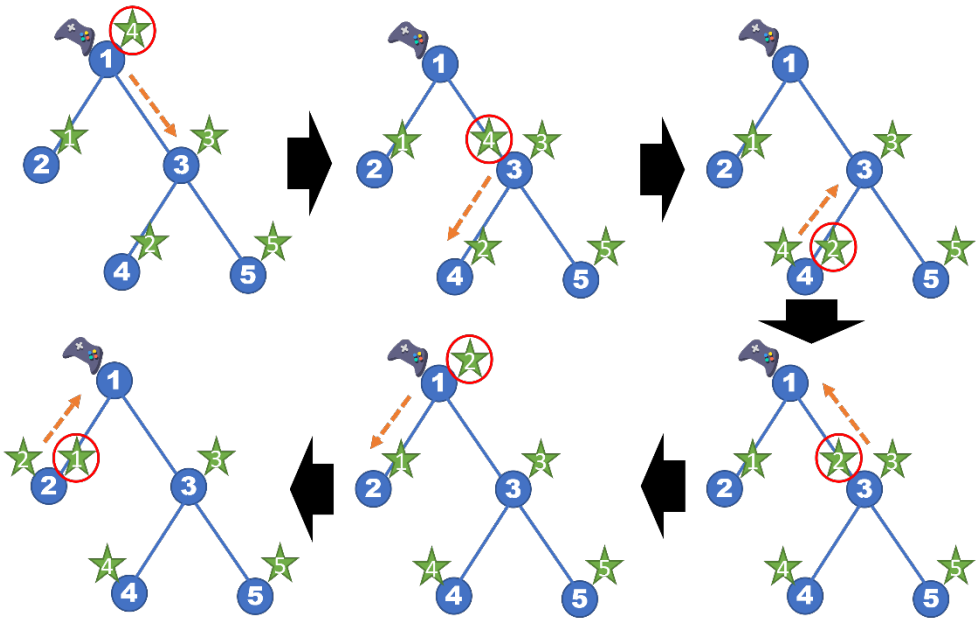
```
3 5
3 7
5 6
```

Output Example 3

```
0
```

Example Explanation:

In Input Example 1, Zisk can:
Pick up the toy in room 1 → Make two moves to room 4 → Swap the toy in room 4 → Make three moves to room 2 → Swap the toy in room 2 → Make one move to room 1 → Put down the toy in his hand
Then he can finish the task in six moves.



In Input Example 2, Zisk can:
Make one move to room 3 → Pick up the toy in room 3 → Make one move to room 5 → Swap the toy in room 5 → Make one move to room 3 → Put down the toy in his hand → Make two moves to room 2 → Pick up the toy in room 2 → Make one move to room 4 → Swap the toy in room 4 → Make one move to room 7 → Swap the toy in room 7 → Make one move to room 4 → Swap the toy in room 4 → Make one move to room 6 → Swap the toy in room 6 → Make two moves to room 2 → Put down the toy in his hand → Make one move to room 1
Then he can finish the task in six moves. Notice that even though the toys have been tidied up, Zisk still needs to go back to the original room (room 1) to keep playing his game.

In Input Example 3, since Zisk's toys have already been tidied up, he doesn't need any move. He can just stay in place and play his game happily.

問題 9 – 凌亂不堪的玩具 (Messy Toys)

(25 分)

問題敍述

天才兒童 Zisk 非常熱愛資訊科學，也因此他要求自己的家必須形如一棵資訊科學中 N個節點的樹，也就是說，恰有 $N-1$ 條通路連接著他家中的 N 個房間，每條通路雙向地連接著兩個房間，且滿足任意兩個房間都能夠經過唯一的一連串通路以互相來往。

不過，天才兒童也是需要放鬆玩耍的，這天 Zisk 玩遊戲玩到一半時突然發現他的玩具們散落一地，原本這些玩具每個都恰好有一個自己歸屬的獨特房間，而每個房間裡也恰好只擺放著一個玩具。

Zisk 想一一將他們放回原位，**放完之後再回來原地繼續玩還沒玩完的遊戲**，但他又很懶得移動，因此在那之前，熱愛資訊科學的他想到了一個問題，如果他只會進行以下四種操作的話，他能**移動**的最少步數是多少呢？

而這四種操作分別是：
1. 在手上沒有任何玩具的情況下，撿起所在房間的玩具
2. 在手上有玩具、所在房間沒有玩具的情況下，將其放置在所在房間。
3. 在手上有玩具、所在房間也有玩具的情況下，將手上的玩具與所在房間的玩具交換。
4. 透過一條連接著所在房間的通路走到另一個房間。

也就是說，Zisk 希望最小化使用操作 4 的次數，來將所有玩具歸回原位，並回到原地繼續玩還沒玩完的遊戲。

Zisk 知道你也很熱愛資訊科學，因此他決定邀請你決鬥，來較量看看誰可以比較早解出這道題目。

輸入格式

首行輸入兩個正整數 N, X，代表 Zisk 家中的地形圖是一棵 N 個點的樹，且 Zisk 與他的遊戲機在房間 X。
接下來一行以空格隔開的 N 個數字 $p_1, \ldots\ldots, p_N$，代表房間 i 上擺放著必須歸位到房間 p_i 的玩具。
接下來 $N-1$ 行，第 i 行兩個以空格隔開的正整數u_i, v_i，代表房間 u_i 和房間 v_i 之間有一條通路。

輸出格式

輸出一個整數，代表 Zisk 將所有玩具歸回原位，並回到原地繼續玩還沒玩完的遊戲所需要移動的最少步數。

資料範圍

- $1 \leq N \leq 600$
- $1 \leq X, u_i, v_i \leq N$
- $p_1 \sim p_N$ 是一個 $1 \sim N$ 的排列
- 保證 Zisk 家中的地形圖是一棵合法 N 個點的樹

輸入範例 1

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

輸出範例 1

```
6
```

輸入範例 2

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

輸出範例 2

```
12
```

輸入範例 3

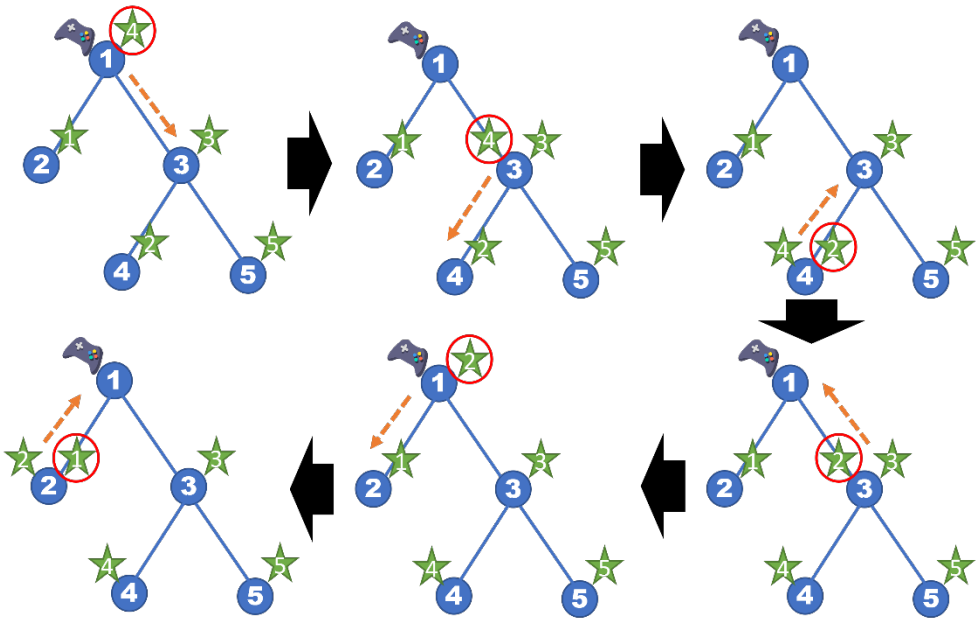
```
8 6
1 2 3 4 5 6 7 8
1 2
1 3
2 4
2 8
3 5
3 7
5 6
```

輸出範例 3

```
0
```

範例說明

在輸入範例 1，Zisk 可以：
先拿起房間 1 的玩具 → 走兩步到房間 4 → 交換房間 4 的玩具 → 走三步到房間 2 → 交換房間 2 的玩具 → 走一步到房間 1 → 放下手上的玩具
便可以花六步完成任務。



在輸入範例 2，Zisk 可以：
先走一步到房間 3 → 拿起房間 3 的玩具 → 走一步到房間 5 → 交換房間 5 的玩具 → 走一步到房間 3 → 放下手上的玩具 → 走兩步到房間 2 → 拿起房間 2 的玩具 → 走一步到房間 4 → 交換房間 4 的玩具 → 走一步到房間 7 → 交換房間 7 的玩具 → 走一步到房間 4 → 交換房間 4 的玩具 → 走一步到房間 6 → 交換房間 6 的玩具 → 走兩步到房間 2 → 放下手上的玩具 → 走一步到房間 1
便可以花十二步完成任務，注意到即使玩具已經整理完畢 Zisk 還是得走回原地（房間 1）繼續玩遊戲。

在輸入範例 3，因為 Zisk 的玩具們都已經歸位了，所以他不需要任何移動，開心待在原地繼續玩遊戲就好。