Containers ( CTAIN )

**Giới hạn**

* **Thời gian:** 5.0s
* **Bộ nhớ:** 1536MB
* **Mã nguồn:** 50000 bytes

We are given n containers, where 1 <= n <= 4. At the beginning all of them are full of water. The liter capacity of the i -th container is a natural number o isatisfying inequalities 1 <= o i<= 49.   
Three kinds of moves can be made:

1. Pouring the whole content of one container into another. This move can be made unless there is too little room in the second container.
2. Filling up one container with part of the water from another one.
3. Pouring away the whole content of one container into a drain.

Task

Write a program that for each test case:

* Reads the number of containers n , the capacity of each container and the requested final amount of water in each container.
* Verifies, whether there exist a series of moves which leads to the requested final situation, and if there is one, the program computes the minimal number of moves leading to the requested situation,
* Writes the result. The result should be the minimal number of moves leading to the requested final situation, or one word "NO" if there is no such a sequence of moves.

Input

One integer in the first line, stating the number of test cases, followed by a blank line. There will be not more than 20 tests.

For each test case, at the first line, one positive integer n is written, n <= 4, this is the number of containers. There are n positive integers written in the second line. These are the capacities of the containers (the i -th integer o idenotes the capacity if the i -th  container,1 <= o i<= 49). In the third line there are written n numbers. These are the requested final volumes of water in the containers (the i -th integer w idenotes the requested final volume of water in the i -th container, 0 <= w i<= o i). All integers in the second and the third line are separated by single spaces.

The test cases will be separated by a single blank line.

Output

For each test case : write one integer - the minimal number of moves which lead to the requested final situation or write only one word "NO" if it is not possible to reach the requested final situation making only allowed moves.

Example

Input:

2

3

3 5 5

0 0 4

2

20 25

10 16

Output:

6

NO

* **Người up:** thanhvy
* **Điểm:** 0.79
* **Ngôn ngữ cho phép:**
* **Nguồn bài:** 3rd Polish Olympiad in Informatics, stage 1

Roads ( ROADS )

**Giới hạn**

* **Thời gian:** 7.0s
* **Bộ nhớ:** 1536MB
* **Mã nguồn:** 50000 bytes

Có N thành phố 1..N nối bởi các con đường một chiều. Mỗi con đường có hai giá trị: độ dài và chi phí phải trả để đi qua. Bob ở thành phố 1. Bạn hãy giúp Bob tìm đường đi ngắn nhất đến thành phố N, biết rằng Bob chỉ có số tiền có hạn là K mà thôi.

Dữ liệu

Dòng đầu tiên ghi t là số test. Với mỗi test, dòng đầu ghi K (0 ≤ K ≤ 10000). Dòng 2 ghi N, 2 ≤ N ≤ 100. Dòng 3 ghi R, 1 ≤ R ≤ 10000 là số đường nối. Mỗi dòng trong N dòng sau ghi 4 số nguyên S, D, L, T mô tả một con đường nối giữa S và D với độ dài L ( 1 ≤ L ≤ 100) và chi phí T (0 ≤ T ≤ 100). Lưu ý có thể có nhiều con đường nối giữa hai thành phố.

Kết quả

Với mỗi test, in ra độ dài đường đi ngắn nhất từ 1 đến N mà tổng chi phí không quá K. Nếu không tồn tại, in ra -1.

Ví dụ

Dữ liệu

2

5

6

7

1 2 2 3

2 4 3 3

3 4 2 4

1 3 4 1

4 6 2 1

3 5 2 0

5 4 3 2

0

4

4

1 4 5 2

1 2 1 0

2 3 1 1

3 4 1 0

Kết quả

11

-1

* **Người up:** paulmcvn
* **Điểm:** 0.13
* **Ngôn ngữ cho phép:**
* **Nguồn bài:** Central European Olympiad in Informatics '98

Đổi chỗ ( NK05EOPR )

**Giới hạn**

* **Thời gian:** 18.0s
* **Bộ nhớ:** 1536MB
* **Mã nguồn:** 50000 bytes

Cho một dãy số là một hoán vị của 12 số tự nhiên đầu tiên (từ 0 đến 11). Giả sử số 0 ở vị trí thứ i trong dãy số (vị trí được đánh số từ 0 đến 11, từ trái sang phải) thì bạn có thể đổi chỗ số 0 với số ở vị trí thứ j nếu thỏa mãn cả hai điều kiện sau:

* | i – j | = d k, với k=1..3 và (d 1,d 2,d 3,d 4)=(1;3;6;12)
* [i/d k+1]=[j/d k+1], với [] là hàm phần nguyên

Bạn hãy tìm số phép đổi chỗ ít nhất để có thể sắp xếp dãy số theo thứ tự tăng dần

Dữ liệu vào

Dòng đầu tiên là một số nguyên t cho biết số lượng test (t<=20)

Mỗi bộ test bao gồm một dòng là dãy bao gồm các số từ 0 đến 11, mỗi số ngăn cách bởi một khoảng trắng.

Biết rằng mỗi dãy số cho trước luôn luôn có thể sắp xếp tăng dần bằng phép đổi chỗ đã quy định

Kết qủa

Với mỗi bộ test, in ra số phép đổi chỗ ít nhất để sắp xếp dãy số đã cho theo thứ tự tăng dần

Ví dụ

Dữ liệu mẫu

2

1 10 2 3 0 5 7 4 8 6 9 11

6 4 1 0 3 5 9 7 2 10 11 8

Kết qủa

8

9

* **Người up:** paulmcvn
* **Điểm:** 1.67
* **Ngôn ngữ cho phép:**
* **Nguồn bài:** Từ acm.uva.es

Trật tự ( NK05ORDR )

**Giới hạn**

* **Thời gian:** 0.374s
* **Bộ nhớ:** 1536MB
* **Mã nguồn:** 50000 bytes

Xét các số nguyên từ 1 đế N. Các số này được sắp xếp theo thứ tự từ điển. Ví dụ với N=11, ta có dãy số sau khi sắp xếp là 1, 10, 11, 2, 3, 4, 5, 6, 7, 8, 9.

Ký hiệu Q N,Klà vị trí của số K trong dãy được sắp xếp theo cách nói trên. Ví dụ Q 11,2=4 Cho các số nguyên K và M. Hãy tìm số nguyên N nhỏ nhất thỏa mãn Q N,K=M

Dữ liệu vào

Dòng đầu tiên chứa số nguyên t cho biết số bộ test.

Mỗi bộ test bao gồm 1 dòng duy nhất chứa 2 số nguyên K và M (1<=K,M<=10 9)

Kết qủa

Với mỗi bộ test xuất ra số N, hoặc 0 nếu không tồn tại N

Ví dụ

Dữ liệu mẫu

1

2 4

Kết qủa

11

* **Người up:** paulmcvn
* **Điểm:** 0.59
* **Ngôn ngữ cho phép:**
* **Nguồn bài:** NEERC 2002

Bốc sỏi ( NK05MNIM )

**Giới hạn**

* **Thời gian:** 0.46s
* **Bộ nhớ:** 1536MB
* **Mã nguồn:** 50000 bytes

Hai bạn Nam và Mai cùng chơi một trò chơi với n đống sỏi. Luật chơi như sau:

* Hai bạn sẽ lần lượt đi. Bạn Mai là người đi trước
* Trong mỗi lượt đi, bạn đi sẽ được quyền bốc một số sỏi bất kỳ từ một đống nhất định và phải bốc tối thiểu là 1 viên sỏi.
* Bạn nào bốc phải viên sỏi cuối cùng là người thua cuộc

Bạn hãy giúp Mai xác định xem bạn ấy có thể thắng được trong trò chơi hay không

Dữ liệu vào

Dòng đầu tiên chứa một nguyên t là số bộ test. Các dòng sau là t bộ test.

Mỗi bộ test bao gồm:

* Dòng đầu tiên chứa một số nguyên n (n<=100) là số đống sỏi
* Dòng thứ hai gồm n số nguyên a 1, a 2, a 3,... , a n, ngăn cách nhau bởi một khoảng trắng. Số nguyên ai cho biết số lượng viên sỏi có trong đống thứ i (1<=a i<=100)

Kết qủa

Với mỗi bộ test, in ra 1 nếu bạn Mai thắng, -1 nếu bạn Mai thua

Ví dụ

Dữ liệu mẫu

2

4

30 4 19 75

3

1 4 5

Kết qủa

1

-1

* **Người up:** paulmcvn
* **Điểm:** 0.23
* **Ngôn ngữ cho phép:**
* **Nguồn bài:** Không rõ

Sa mạc ( NK05DSRT )

**Giới hạn**

* **Thời gian:** 0.469s
* **Bộ nhớ:** 1536MB
* **Mã nguồn:** 50000 bytes

Bờm vô tình bị lạc vào trong 1 ốc đảo có 1 bộ tộc thổ dân sinh sống trong 1 lần đi qua sa mạc. Bờm muốn thoát khỏi sa mạc để về nhà. Người thổ dân đã cho anh một bản đồ vùng sa mạc này.

Sa mạc gồm N ốc đảo, M đường đi an toàn nối với nhau và tại mỗi ốc đảo lại có 1 hồ chứa nước rất lớn và nước chứa trong các hồ này không bao giờ cạn. Tuy nhiên hiện tại, không có nước trong các hồ.

Giả sử: Bờm đang ở ốc đảo 1, về để về đến nhà thì Bờm phải đi đến ốc đảo N. Người thổ dân cho biết rằng tại vùng sa mạc này, nếu Bờm đi đoạn đường có độ dài là L, thì Bờm phải mang uống đủ lượng nước là L, bằng không Bờm sẽ chết. Từ đó, người thổ dân đã chỉ cho Bờm cách có thể về nhà: Bờm phải vận chuyển nước từ ốc đảo 1 đển tích trữ trong các hồ tại những ốc đảo khác bằng các con đường an toàn đã có, từ đó anh có thể về nhà. Tuy nhiên, lại có 1 khó khăn khác là: trong bất cứ thời gian nào, Bờm không thể mang lượng nước quá C (do thể lực có hạn ^\_^).

Yêu cầu: Hãy tìm cách giúp Bờm về nhà nhưng đồng thời sử dụng ít nước nhất (do trong sa mạc, nước rất quí !!!!)

Dữ liệu vào

Dòng đầu tiên gồm một số nguyên t cho biết số lượng test, mỗi test có dạng như sau:

* Dòng đầu tiên là 3 số nguyên N M C ngăn cách nhau bởi khoảng trắng
* M dòng tiếp theo mỗi dòng gồm 3 số nguyên I J L với y nghĩa có đường đi từ ốc đảo I đến ốc đảo J và ngược lại là an toàn và có độ dài L.

Kết qủa

Với mỗi bộ test xuất ra đúng 1 số nguyên chỉ lượng nước ít nhất cần dùng.

Giới hạn

* 1 <= N, M, C <= 100
* 1 <= L <= 30,000

Ví dụ

Dữ liệu mẫu

1

9 10 25

1 2 3

2 3 12

3 4 4

3 5 9

4 9 13

5 9 5

2 6 10

6 7 10

7 8 10

8 9 10

Kết qủa

65

Giải thích

Mang 25 nước từ 1 đến 2 sau đó lại quay về 1. Do đó, tại 2 có 19 nước. (Bờm đã uống hết (3 + 3) nước trong lần đi và về, (19 = 25 – 3 – 3)).

Lặp lại như thế 1 lần nữa, Bờm đã mang đến 2 thêm 19 nước. Sau đó lại từ 1, Bờm mang theo 15 nước đến 2. Vậy khi đến 2, Bờm có tại đây (19+19+12 = 50) nước.

Tiếp theo, Bờm lại mang 25 nước đi từ 2 đến 3, rồi quay về 2, như vậy, tại 3 có (25 – 12 – 12 = 1) nước. Từ 2, Bờm mang 25 nước còn lại đi đến 3. Tại 3, Bờm có được (1+(25-12) = 14) nước.

Cuối cùng, Bờm mang 14 nước đi đến 5 rồi đến 9. Vậy là Bờm đã thoát khỏi sa mạc.

* **Người up:** paulmcvn
* **Điểm:** 0.30
* **Ngôn ngữ cho phép:**
* **Nguồn bài:** Lê Thành Trung - 3rd VNOPSC

Lazy Cows ( LAZYCOWS )

**Giới hạn**

* **Thời gian:** 9.0s
* **Bộ nhớ:** 1536MB
* **Mã nguồn:** 50000 bytes

Cho mô hình bãi cỏ có dạng bảng chữ nhật 2xB (1 <= B <= 15,000,000), trong đó một số ô có con bò đang ăn cỏ. Có tất cả N con bò (1 <= N <= 1000) trên bãi cỏ. Ví dụ:

-------------------------------------------------------

| | cow | | | | cow | cow | cow | cow |

-------------------------------------------------------

| | cow | cow | cow | | | | | |

-------------------------------------------------------

Bác John muốn dùng K (1 <= K <= N) cái chuồng hình chữ nhật để che phủ đàn bò, sao cho tổng diện tích được phủ là nhỏ nhất. Không có hai chuồng nào được nằm đè lên nhau. Hiển nhiên, các chuồng phải che phủ hết các ô có bò.

Ví dụ, trong hình trên nếu K=2, lời giải tối ưu bao gồm một chuồng kích thước 2x3 và một chuồng kích thước 1x4. Tổng diện tích được phủ là 10.

Dữ liệu

Dòng đầu tiên chứa một số nguyên t là số bộ test. Mỗi bộ test có dạng:

* Dòng 1: N, K, B
* Dòng 2..N+1: chức tọa độ của các con bò, trong phạm vi (1,1) đến (2,B). Không có ô nào chứa hơn 1 con bò.

Kết quả

Với mỗi test, in ra tổng diện tích nhỏ nhất cần phủ.

Ví dụ

Dữ liệu

1

8 2 9

1 2

1 6

1 7

1 8

1 9

2 2

2 3

2 4

Kết quả

10

* **Người up:** paulmcvn
* **Điểm:** 0.59
* **Ngôn ngữ cho phép:**
* **Nguồn bài:** US Open International 2005 Gold Division

Boxes ( BOXES )

**Giới hạn**

* **Thời gian:** 7.273s
* **Bộ nhớ:** 1536MB
* **Mã nguồn:** 50000 bytes

Có n cái hộp xếp theo vòng tròn đánh số 1..n (1 ≤ n ≤ 1000) theo chiều kim đồng hồ. Mỗi hộp chứa một số quả bóng, tổng số quả bóng không quá n.

Cần dịch chuyển các quả bóng sao cho mỗi hộp không chứa quá 1 quả. Mỗi bước, ta có thể di chuyển một quả bóng từ một hộp sang một trong hai hộp bên cạnh.

Tính số bước di chuyển ít nhất.

Dữ liệu

Dòng đầu tiên chứa t là số bộ test (t ≤ 20). Mỗi bộ test có dạng:

* Dòng đầu tiên: n - số hộp.
* Dòng thứ hai: n số nguyên không âm là số quả bóng trong các hộp

Kết quả

Với mỗi bộ test in ra số bước ít nhất cần thiết.

Ví dụ

Dữ liệu

1

12

0 0 2 4 3 1 0 0 0 0 0 1

Kết quả

19

* **Người up:** paulmcvn
* **Điểm:** 0.87
* **Ngôn ngữ cho phép:**
* **Nguồn bài:** III Polish Olympiad in Informatics, stage 3

Count maximum matrices ( MATRIX )

**Giới hạn**

* **Thời gian:** 2.538s
* **Bộ nhớ:** 1536MB
* **Mã nguồn:** 50000 bytes

You are given a matrix A of M rows and N columns, consisting of numbers 0 and 1. For a rectangle in A (sides >= 1), X1 is the number of ones on its sides, X0 is the number of zeros on its sides, and its value is defined as X1 – X0. Let us consider W, the maximum value taken over submatrices of A, and S, the number of submatrices with value W. Your task is to find W and S.

Input

The first line of input contains the number of testcases t (t <= 15). The first line of each testcase contains the numbers M, N (1 <= M, N <= 200) Then M lines follow. In each line, there are N numbers 0 or 1.

Output

For each testcase, you should output a single line with numbers W and S.

Example

Input:

1

5 6

1 1 1 1 1 1

1 0 0 0 0 1

1 0 0 0 0 1

1 0 0 0 0 1

1 1 1 1 1 1

Output:

18 1

* **Người up:** beo\_chay\_so
* **Điểm:** 0.75
* **Ngôn ngữ cho phép:**

Another Assignment Problem ( ASSIGN4 )

**Giới hạn**

* **Thời gian:** 0.469s
* **Bộ nhớ:** 1536MB
* **Mã nguồn:** 50000 bytes

Assume that you are a manager and there are m types of worker (numbered from 1 to m) and n types of task (numbered from 1 to n). There are a(i) workers of type #i and b(j) postitions for task #j. C(i, j) is the cost of hiring a worker of type #i to do the task of type #j. Your job is to minimize the cost of hiring workers to fill all the positions given that the total number of workers is equal to the total number of positions.

Input

The first line of input contains the number of test cases nTest (1<= nTest <= 10). Each test case contains:

* The first line contains the number of worker types - m and number of task types - n.
* The second line contains m positive integers: a(1), a(2), ..., a(m).
* The third line contains n positive integers: b(1), b(2), ..., b(n).
* Each of the next m lines contains n integers describing matrix C(i, j).

Notes:

1 <= m, n <= 200;

1 <= a(i), b(i) <= 30000;

1 <= C(i, j) <= 10000.

Sum of a(i) equals to sum of b(j).

Output

For each test case write the minimum cost in a separate line (it will fit in a signed 32-bit integer).

Example

Input:

2

3 4

3 6 7

2 5 1 8

1 2 3 4

8 7 6 5

9 12 10 11

4 4

1 3 5 7

2 4 2 8

1 4 7 3

4 7 5 3

5 7 8 3

5 3 6 8

Output:

110

54

* **Người up:** dtmp
* **Điểm:** 1.14
* **Ngôn ngữ cho phép:**
* **Nguồn bài:** Tran Quang Khai

Nhân 1 ( MULONE )

**Giới hạn**

* **Thời gian:** 0.759s
* **Bộ nhớ:** 1536MB
* **Mã nguồn:** 50000 bytes

Cho số S = 111...11 (n chữ số 1, hệ thập phân), tính S 2.

Input

- Dòng đầu tiên: số lượng test k.

- k dòng tiếp, mỗi dòng ghi số n - số lượng chữ số 1 của S. (1 <= n <= 1000000)

Output

- Với mỗi test ghi kết quả trên 1 dòng.

Example

**Input:**

2

1

2

**Output:**

1

121

* **Người up:** dtmp
* **Điểm:** 0.06
* **Ngôn ngữ cho phép:**

Mountain Walking ( MTWALK )

**Giới hạn**

* **Thời gian:** 0.164s
* **Bộ nhớ:** 1536MB
* **Mã nguồn:** 50000 bytes

Cho một bản đồ kích thước NxN (2 <= N <= 100), mỗi ô mang giá trị là độ cao của ô đó (0 <= độ cao <= 110). Bác John và bò Bessie đang ở ô trên trái (dòng 1, cột 1) và muốn đi đến cabin (dòng N, cột N). Họ có thể đi sang phải, trái, lên trên và xuống dưới nhưng không thể đi theo đường chéo. Hãy giúp bác John và bò Bessie tìm đường đi sao cho chênh lệch giữa điểm cao nhất và thấp nhất trên đường đi là nhỏ nhất.

Dữ liệu

* Dòng 1: N
* Dòng 2..N+1: Mỗi dòng chứa N số nguyên, mỗi số cho biết cao độ của một ô.

Kết quả

Một số nguyên là chênh lệch cao độ nhỏ nhất.

Ví dụ

Dữ liệu

5

1 1 3 6 8

1 2 2 5 5

4 4 0 3 3

8 0 2 3 4

4 3 0 2 1

Kết quả

2

* **Người up:** paulmcvn
* **Điểm:** 0.12
* **Ngôn ngữ cho phép:**
* **Nguồn bài:** USACO 2003 US Open

Duyệt binh ( PYRAMID2 )

**Giới hạn**

* **Thời gian:** 0.819s
* **Bộ nhớ:** 1536MB
* **Mã nguồn:** 50000 bytes

Vào năm 1945, Liên Xô đang đánh nhau với phát xít Đức hết sức ác liệt. Hàng triệu thanh niên Liên Xô phải lên đường nhập ngũ. Một cuộc duyệt binh diễn ra, các tân binh không biết đứng quay mặt về bên nào liền xếp tùy ý, vị tổng chỉ huy thấy thể liền ra lệnh: “Nếu hai tân binh liên tiếp và đối mặt với nhau thì ngay lập phải quay ngược lại(180 độ), động tác này diễn ra trong vòng 1s!”. Người tổng chỉ huy muốn biết sau bao lâu thì thì đội hình sẽ ngừng quay?

Input

Dòng đầu ghi số nguyên N là số tân binh.   
Dòng thứ hai gồm đúng N kí tự ‘<’, ‘>’ thể hiện cách đứng của các tân binh. Nếu hai tân binh liên tiếp quay mặt vào nhau thì sẽ được biểu diễn bởi ‘><’. ( 1 ≤ n ≤ 1000000 ).

Output

Gồm một số duy nhất ghi thời gian để đội hình ngừng quay.

Example

Input:

4

<><>

Output:

1

Chú thích :   
Tại thời điểm 0: <><>   
Tại thời điểm 1: <<>>

* **Người up:** hard7771988
* **Điểm:** 0.13
* **Ngôn ngữ cho phép:**
* **Nguồn bài:** Hiếu add ðấy nhé , kô chú Tư lại kêu ca ( cái account tớ bị lỗi hay s

Truyền tin ( MESSAGE )

**Giới hạn**

* **Thời gian:** 0.086s
* **Bộ nhớ:** 1536MB
* **Mã nguồn:** 50000 bytes

Một lớp gồm N học sinh, mỗi học sinh cho biết những bạn mà học sinh đó có thể liên lạc được (chú ý liên lạc này là liên lạc một chiều : u có thể gửi tin tới v nhưng v thì chưa chắc đã có thể gửi tin tới u).

Thầy chủ nhiệm đang có một thông tin rất quan trọng cần thông báo tới tất cả các học sinh. Để tiết kiệm thời gian, thầy chỉ nhắn tin tới 1 số học sinh rồi sau đó nhờ các học sinh này nhắn lại cho tất cả các bạn mà các học sinh đó có thể liên lạc được, và cứ lần lượt như thế làm sao cho tất cả các học sinh trong lớp đều nhận được tin .

Hãy tìm một số ít nhất các học sinh mà thầy chủ nhiệm cần nhắn.

Input

- Dòng đầu là N, M (N <= 800, M là số lượng liên lạc 1 chiều)

- Một số dòng tiếp theo mỗi dòng gồm 2 số u , v cho biết học sinh u có thể gửi tin tới học sinh v

Output

- Gồm 1 dòng ghi số học sinh cần thầy nhắn tin.

Example

Input:

12 15

1 3

3 6

6 1

6 8

8 12

12 9

9 6

2 4

4 5

5 2

4 6

7 10

10 11

11 7

10 9

Output:

2

Chọn các học sinh 7 và 2.

* **Người up:** dtmp
* **Điểm:** 0.07
* **Ngôn ngữ cho phép:**

Dãy số Catalan ( CATALAN )

**Giới hạn**

* **Thời gian:** 0.167s
* **Bộ nhớ:** 1536MB
* **Mã nguồn:** 50000 bytes

Cho số nguyên dương N, dãy Catalan cấp n là dãy C(1), C(2) … C(2n+1) gồm các số nguyên không âm thoả mãn : C(1) = C(2n+1) = 0 với i bất kì 1 ≤ i ≤ 2n thì C(i), C(i+1) hơn kém nhau 1 đơn vị.

Với mỗi n ta sắp xếp các dãy Catalan theo thứ tự từ điển, đánh số từ 1 trở đi . Yêu cầu :

1.Cho một dãy Catalan, hãy tìm thứ tự của dãy.

2.Cho số nguyên dương k hãy tìm dãy có thứ tự k

Input

- Dòng đầu ghi n. (n <= 15)

- Dòng hai ghi một dãy Catalan cấp n

- Dòng 3 ghi một số nguyên dương k (k có thể rất lớn nhưng đảm bảo luôn có nghiệm)

Output

- Dòng 1 ghi số thứ tự dãy ở dòng 2 INPUT

- Dòng 2 ghi dãy ứng với số thứ tự

Example

Input:

4

0 1 2 3 2 1 2 1 0

12

Output:

12

0 1 2 3 2 1 2 1 0

* **Người up:** dtmp
* **Điểm:** 0.14
* **Ngôn ngữ cho phép:**

Chữ số tận cùng khác 0 ( TCDFZ )

**Giới hạn**

* **Thời gian:** 0.098s
* **Bộ nhớ:** 1536MB
* **Mã nguồn:** 50000 bytes

Cho số tự nhiên n (n <= 10^9). Hãy tìm chữ số tận cùng khác 0 của n!

Input

- Dòng duy nhất ghi số N.

Output

- Gồm 1 dòng duy nhất ghi kết quả tìm được.

Example

**Input:**

5

**Output:**

2

* **Người up:** dtmp
* **Điểm:** 0.17
* **Ngôn ngữ cho phép:**

Thang máy vũ trụ ( ELEVATOR )

**Giới hạn**

* **Thời gian:** 1.013s
* **Bộ nhớ:** 1536MB
* **Mã nguồn:** 50000 bytes

Những con bò muốn đi vào vũ trụ! Chúng muốn đến được quỹ đạo bằng cách xây một kiểu thang máy: một cái tháp khổng lồ làm bằng các khối chồng lên nhau. Chúng có K (1 ≤ K ≤ 400) loại khối có thể xây tháp. Mỗi khối loại i có chiều cao h\_i (1 ≤ h\_i ≤ 100) và có số lượng c\_i (1 ≤ c\_i ≤ 10). Do khả năng bị phá hủy bởi các tia vũ trụ, không có phần nào của khối loại i có thể vượt qua độ cao a\_i (1 ≤ a\_i ≤ 40000).

Giúp những con bò xây thang máy cao nhất có thể bằng cách chồng các khối lên nhau theo luật trên.

Input

\* Dòng 1: Một số nguyên: K

\* Dòng 2..K+1: Mỗi dòng chứa 3 số nguyên được phân cách bởi khoảng trắng: h\_i, a\_i, và c\_i. Dòng i+1 miêu tả loaị khối i.

Output

\* Dòng 1: Một số nguyên H, chỉ độ cao lớn nhất của tháp có thể xây được.

Example

Input:

3

7 40 3

5 23 8

2 52 6

Output:

48

GIẢI THÍCH:

Từ dưới lên: 3 khối loại 2, 3 khối loại 1, 6 khối loại 3. Chồng 4 khối loại 2 & 3 loại 1 không hợp lệ vì đỉnh của khối loại 1 vượt quá độ cao 40.

* **Người up:** dtmp
* **Điểm:** 0.13
* **Ngôn ngữ cho phép:**
* **Nguồn bài:** USACO 3/2005

Dãy ngoặc ( BRACKET )

**Giới hạn**

* **Thời gian:** 0.206s
* **Bộ nhớ:** 1536MB
* **Mã nguồn:** 50000 bytes

Người ta định nghĩa đệ qui dãy ngoặc và cấp của dãy như sau:

+) Xâu rỗng được gọi là dãy ngoặc cấp 0.

+) Nếu S là xâu ngoặc cấp k thì (S) là xâu ngoặc cấp k+1.

+) Nếu A, B là các dãy ngoặc thì S = AB là một dãy ngoặc với cấp bằng số lớn hơn trong cấp của A và B.

Định nghĩa này chỉ áp dụng cho những xâu sinh ra theo qui tắc đệ qui trên.

Cho 2 số nguyên dương N và k, gọi S là tập các dãy ngoặc cấp k độ dài N.

1. Cho biết S có bao nhiêu phần tử.

2. Cho một dãy ngoặc thuộc, hãy cho biết thứ tự từ điển của dãy này trong tập S.

Input

- Dòng đầu ghi 2 số N, k (N chẵn, N <= 60, k <= n/2 ).

- Dòng hai ghi 1 xâu ngoặc cấp k độ dài N.

Output

Gồm hai dòng, mỗi dòng trả lời 1 yêu cầu theo thứ tự trên.

Example

Input:

6 2

(())()

Output:

3

2

* **Người up:** dtmp
* **Điểm:** 0.31
* **Ngôn ngữ cho phép:**

Quan hệ ( COND )

**Giới hạn**

* **Thời gian:** 0.134s
* **Bộ nhớ:** 1536MB
* **Mã nguồn:** 50000 bytes

Xét một tập N đối tượng có thể so sánh được (2<=n<=10). Giữa 2 đối tượng a và b có thể tồn tại 1 trong 3 quan hệ phân loại:

a = b; a < b; a > b;

Như vậy, với 3 đối tượng (a, b, c) có thể tồn tại 13 quan hệ phân loại như sau:

a = b = c; a = b < c; c < a = b; a < b = c   
b = c < a; a = c < b; b < a = c; a < b < c   
a < c < b; b < a < c; b < c < a; c < a < b   
c < b < a;

Cho số n, hãy xác định số lượng quan hệ phân loại khác nhau.

Input

Gồm nhiều số n. Mỗi số trên 1 dòng. Kết thúc file là -1.

Output

Với mỗi n, đưa ra số lượng quan hệ phân loại tìm được, mỗi số trên 1 dòng (không có dòng trống).

Example

Input:

2

3

-1

Output:

3

13

* **Người up:** dtmp
* **Điểm:** 0.13
* **Ngôn ngữ cho phép:**

Hai đường đi ( HIWAY )

**Giới hạn**

* **Thời gian:** 0.188s
* **Bộ nhớ:** 1536MB
* **Mã nguồn:** 50000 bytes

Một mạng giao thông gồm N nút giao thông, và có M đường hai chiều nối một số cặp nút, thông tin về một đường gồm ba số nguyên dương u, v là tên hai nút đầu mút của đường, và l là độ dài đoạn đường đó. Biết rằng hai nút giao thông bất kì có không quá 1 đường hai chiều nhận chúng làm hai đầu mút.   
Cho hai nút giao thông s và f, hãy tìm hai đường đi nối giữa s với f sao cho hai trên hai đường không có cạnh nào được đi qua hai lần và tổng độ dài 2 đường đi là nhỏ nhất.

Input

- Dòng đầu ghi N, M (N ≤ 100)   
- Dòng thứ 2 ghi hai số s, f.   
- M dòng tiếp theo, mỗi dòng mô tả một đường gồm ba số nguyên dương u, v, l.

Output

- Dòng đầu ghi T là tổng độ dài nhỏ nhất tìm được hoặc -1 nếu không tìm được.   
- Nếu tìm được, hai dòng sau, mỗi dòng mô tả một đường đi gồm: số đầu là số nút trên đường đi này, tiếp theo là dãy các nút trên đường đi bắt đầu từ s, kết thúc tại f.

Chú ý: Phạm vi tính toán trong vòng Longint.

Example

Input:

5 8

1 5

1 2 1

1 4 8

2 3 5

2 4 1

3 5 1

4 3 8

4 5 1

1 3 1

Output:

5

3 1 3 5

4 1 2 4 5

* **Người up:** dtmp
* **Điểm:** 0.18
* **Ngôn ngữ cho phép:**

© VNOI Team 2015

Trao đổi thông tin ( KWAY )

**Giới hạn**

* **Thời gian:** 0.301s
* **Bộ nhớ:** 1536MB
* **Mã nguồn:** 50000 bytes

Cho một mạng thông tin gồm n trạm và m đường nối hai chiều giữa các trạm. Trạm s là trạm chỉ huy, trạm f là trạm điều khiển. Sau một lần bị tin tặc tấn công lấy mất dữ liệu từ trạm chỉ huy chuyển đến trạm điều khiển, chỉ huy mạng quyết định chia thông tin chuyển đi thành k đơn vị thông tin để chuyển theo k đường đến trạm điều khiển. Mà hai đường truyền bất kỳ không được chung bất kỳ một đường nào.  
Hãy tìm cách truyền k đơn vị thông tin sao cho tổng chi phí là nhỏ nhất.

Input

- Dòng đầu là n, m, k, s, f (n ≤ 100).   
- m dòng tiếp là u, v, c cho biết có đường từ u -> v và v -> u với chi phí là c.

Output

- Dòng đầu ghi –1 nếu không thể chuyển k đơn vị thông tin theo cách trên, ngược lại ghi chi phi để chuyển.   
- k dòng tiếp lần lượt ghi cách chuyển của từng đơn vị thông tin. Số đầu là số lượng trạm trên đường truyền, tiếp đó là dãy các trạm trên đường truyền (bắt đầu từ s, kết thúc ở f)

Chú ý: Phạm vi tính toán là Longint.

Example

Input:

8 11 3 1 8

1 2 1

1 4 1

1 5 1

2 3 1

2 4 1

2 7 1

3 8 1

3 6 1

3 5 1

6 8 1

7 8 1

Output:

11

4 1 2 3 8

5 1 5 3 6 8

5 1 4 2 7 8

* **Người up:** dtmp
* **Điểm:** 0.39
* **Ngôn ngữ cho phép:**

Luồng với chi phí nhỏ nhất ( MINCOST )

**Giới hạn**

* **Thời gian:** 1.359s
* **Bộ nhớ:** 1536MB
* **Mã nguồn:** 50000 bytes

Cho một mạng đối xứng có n đỉnh, mỗi cạnh của mạng có một khả năng thông qua và một cước phí vận chuyển nhất định (như nhau theo cả hai chiều). Cho trước một lượng hàng S cần vận chuyển từ đỉnh nguồn (đánh số là s) tới đỉnh đích (đánh số là f). Hãy tìm một phương án vận chuyển, nghĩa là hãy xác định trên mỗi cạnh của mạng cần vận chuyển bao nhiêu hàng, theo chiều nào, sao cho phù hợp với khả năng thông qua của mạng (trên mỗi cạnh lượng hàng vận chuyển không vượt quá khả năng thông qua của cạnh) và vận chuyển được lượng hàng S từ nguồn về đích với tổng chi phí vận chuyển là nhỏ nhất.   
Về mặt toán học, bài toán tìm luồng với chi phí nhỏ nhất có thể diễn đạt như sau:

Cực tiểu hóa hàm chi phí ∑c ijx ijvới điều kiện:

1. ∑(x ij- x ji) với j = 1..n, có giá trị   
   * S nếu i = s
   * 0 nếu i ≠ s; i ≠ n
   * -S nếu i = f
2. 0 ≤ x ij≤ d ijvới mọi cạnh (i, j)

Ở đây đỉnh nguồn được đánh số là s, đỉnh đích là f, c ijlà chi phí vận chuyển một đơn vị hàng trên cạnh (i, j), d ijlà khả năng thông qua của cạnh (i, j); còn x ijlà khối lượng hàng vận chuyển trên cạnh (i, j) cần xác định.

Input

* Dòng đầu là n, m, k, s, f : Số đỉnh, số đường, số đơn vị hàng cần vận chuyển. đỉnh bắt đầu, đỉnh kết thúc
* m dòng tiếp theo mỗi bao gồm u, v, c, d cho biết có đường từ u -> v, v -> u với chi phí là c và khả năng thông qua là d.

Output

* Dòng đầu, nếu không vận chuyển được ghi –1, nếu có ghi tổng chi phí vận chuyển.
* Nếu có nghiệm thì một số dòng tiếp ghi u, v, i cho biết vận chuyển i đơn vị hàng từ trên cạnh u -> v. Kết thúc bằng "0 0 0".

Example

Input:

6 8 5 1 6

1 2 1 2

1 4 3 4

2 3 1 4

2 5 5 2

3 4 2 4

3 6 1 2

4 6 4 1

5 6 6 2

Output:

43

1 2 2

1 4 3

2 5 2

3 6 2

4 3 2

4 6 1

5 6 2

0 0 0

Giới hạn: 

* n <= 100
* d ij<= 30000
* c ij<= 10 9

Phạm vi tính toán là Longint.

* **Người up:** dtmp
* **Điểm:** 0.31
* **Ngôn ngữ cho phép:**

Phân công hoàn thành sớm nhất ( ASSIGN1 )

**Giới hạn**

* **Thời gian:** 0.185s
* **Bộ nhớ:** 1536MB
* **Mã nguồn:** 50000 bytes

Có n người, n việc (1 < n ≤ 200). Người thứ i thực hiện công viêc j mất C[i,j] đơn vị thời gian. Giả sử tất cả bắt đầu vào thời điểm 0, hãy tìm cách bố trí mỗi công việc cho mỗi người sao cho thời điểm hoàn thành công việc là sớm nhất có thể.

Input

- Dòng đầu: N   
- Tiếp theo là ma trận C[i,j]. (thuộc kiểu Integer)

Output

- Ghi thời điểm sớm nhất hoàn thành.

Example

Input:

4

10 10 10 2

10 10 3 10

4 10 10 10

10 5 10 10

Output:

5

* **Người up:** dtmp
* **Điểm:** 0.17
* **Ngôn ngữ cho phép:**

Rải sỏi ( STONE1 )

**Giới hạn**

* **Thời gian:** 0.206s
* **Bộ nhớ:** 1536MB
* **Mã nguồn:** 50000 bytes

Xét trò chơi rải sỏi với một người chơi như sau:

Cho cây T và một đống sỏi gồm K viên. Ở mỗi bước người ta lấy 1 viên sỏi từ đống sỏi và đặt vào một nút lá tuỳ ý. Nếu nút p có r nút lá và tất cả các nút lá đều đã có sỏi thì người ta gom tất cả các viên sỏi ở các nút lá lại, đặt 1 viên ở nút p, xoá các nút lá và trả r - 1 viên sỏi còn lại vào đống sỏi.

Trò chơi kết thúc khi đã đặt được 1 viên sỏi vào nút gốc

Yêu cầu: cho cây T, xác định số viên sỏi tối thiểu cần có để trò chơi có thể kết thúc. Cây có n nút (N <= 400), nút gốc được đánh số 1.

Input

* Dòng đầu: số n.
* Một số dòng tiếp theo, mỗi dòng có dạng: i m i1 i2 ... im. Trong đó m là số nút con của nút i; i1, i2, ..., im: các nút con của nút i.

Output

Số lượng viên sỏi ít nhất cần có.

Example

Input:

7

1 2 2 3

2 2 5 4

3 2 6 7

Output:

3

* **Người up:** dtmp
* **Điểm:** 0.25
* **Ngôn ngữ cho phép:**

Xâu con ( SUBSTR )

**Giới hạn**

* **Thời gian:** 0.451s
* **Bộ nhớ:** 1536MB
* **Mã nguồn:** 50000 bytes

Cho xâu A và xâu B chỉ gồm các chữ cái thường. Xâu B được gọi là xuất hiện tại vị trí i của xâu A nếu: A[i] = B[1], A[i+1] = B[2], ..., A[i+length(B)-1] = B[length(B)].

Hãy tìm tất cả các vị trí mà B xuất hiện trong A.

Input

* Dòng 1: xâu A.
* Dòng 2: xâu B.

Độ dài A, B không quá 1000000.

Output

Ghi ra các vị trí tìm được trên 1 dòng (thứ tự tăng dần). Nếu B không xuất hiện trong A thì bỏ trắng.

Example

Input:

aaaaa

aa

Output:

1 2 3 4

* **Người up:** dtmp
* **Điểm:** 0.06
* **Ngôn ngữ cho phép:**

Các hình chữ nhật ( RECT1 )

**Giới hạn**

* **Thời gian:** 0.077s
* **Bộ nhớ:** 1536MB
* **Mã nguồn:** 50000 bytes

Cho N hình chữ nhật trên mặt phẳng. Các cạnh hình chữ nhật song song với các trục tọa độ. Những hình chữ nhật này có thể gối lên nhau, trùng hoặc là bên trong nhau. Đỉnh của chúng có tọa độ nguyên, hoành độ x không vượt quá xmax và tung độ y không vượt quá ymax.   
Một đoạn thẳng có một đầu là điểm A(0, 0) và đầu kia là điểm B. Điểm B thỏa mãn các điều kiện sau:   
+) Các tọa độ của B là những số nguyên.   
+) Điểm B thuộc đoạn [(0, ymax), (xmax, ymax)] hoặc đoạn [(xmax, 0), (xmax, ymax)].   
Viết chương trình tìm một điểm B sao cho đoạn AB cắt qua nhiều hình chữ nhật nhất. (AB cắt 1 hình chữ nhật khi chúng có ít nhất 1 điểm chung với nhau).

Input

- Dòng đầu chứa 3 số nguyên xmax, ymax (0 < xmax, ymax < 10^9) và N (1 <= N <= 10000).   
- Mỗi dòng trong N dòng tiếp theo chứa 4 số nguyên: x1, y1, x2, y2. (x1, y1) là tọa độ đỉnh trái dưới, (x2, y2) là tọa độ đỉnh phải trên của hình chữ nhật tương ứng.

Output

Dòng duy nhất ghi số lượng lớn nhất các hình chữ nhật cắt được.

Example

Input:

22 14 8

1 8 7 11

18 10 20 12

17 1 19 7

12 2 16 3

16 7 19 9

8 4 12 11

7 4 9 6

10 5 11 6

Output:

5

* **Người up:** dtmp
* **Điểm:** 0.61
* **Ngôn ngữ cho phép:**

Khối lập phương lớn nhất ( MAXCUB )

**Giới hạn**

* **Thời gian:** 0.145s
* **Bộ nhớ:** 1536MB
* **Mã nguồn:** 50000 bytes

Cho một khối lập phương kích thước n chia làm n 3khối lập phương đơn vị. Mỗi khối lập phương đơn vị chứa 1 số nguyên.   
Bạn hãy tìm một khối lập phương con của khối lập phương đã cho sao cho tổng các số trong khối lập phương con đó là lớn nhất.

Input

- Dòng đầu: số lượng test.   
- Tiếp theo là các test, mỗi test gồm: dòng đầu là n. Sau đó n nhóm dòng thể hiện lớp cắt của hình lập phương nhìn từ mặt trước từ gần ra xa, mỗi nhóm gồm n dòng, mỗi dòng gồm n số liệt kê các số trên lớp cắt từ trên xuống dưới, trái qua phải.   
Chú ý: n <= 30. Giá trị của khối lập phương đơn vị thuộc kiểu integer.

Output

Mỗi dòng chứa tổng của khối lập phương con lớn nhất của test tương ứng.

Example

Input:

2

3

0 -1 3

-5 7 4

-8 9 1

-1 -3 -1

2 -1 5

0 -1 3

3 1 -1

1 3 2

1 -2 1

4

1 1 1 1

1 1 1 1

1 1 1 1

1 1 1 1

1 1 1 1

1 1 1 1

1 1 1 1

1 1 1 1

1 1 1 1

1 1 1 1

1 1 1 1

1 1 1 1

1 1 1 1

1 1 1 1

1 1 1 1

1 1 1 1

Output:

27

64

* **Người up:** dtmp
* **Điểm:** 0.23
* **Ngôn ngữ cho phép:**

Số chính phương ( CP )

**Giới hạn**

* **Thời gian:** 1.37s
* **Bộ nhớ:** 1536MB
* **Mã nguồn:** 50000 bytes

John là một người rất đam mê toán học, một lần cậu viết ra một dãy số các chữ số và nhận ra rằng dãy số vừa viết có thể tách thành một số đoạn con liên tiếp, mà mỗi đoạn con tạo thành một số là số chính phương.   
Ví dụ: dãy số 149 có thể tách thành 3 đoạn: 1, 4, 9 -> mỗi đoạn đều là số chính phương hoặc có thể tách thành 2 đoạn 1 và 49.   
John muốn biết là có bao nhiêu cách tách khác nhau (hai cách tách được gọi là khác nhau nếu tồn tại một vị trí tách khác nhau) dãy chữ số mình vừa viết. Điều kiện là các đoạn tách ra không bắt đầu bằng chữ số 0 .

Input

- Dòng đầu là số lượng test: nTest.   
- nTest dòng tiếp theo mỗi dòng ghi ra dãy chữ số mà John viết (độ dài không quá 100).

Output

- Với mỗi test ghi ra số lượng cách tìm được trên 1 dòng.

Example

Input:

1

169

Output:

2

169 -> 169 = 13^2   
169 -> 16 = 4^2 và 9 = 3^2.

* **Người up:** dtmp
* **Điểm:** 0.77
* **Ngôn ngữ cho phép:**

Mưa thiên thạch ( METERAIN )

**Giới hạn**

* **Thời gian:** 0.119s
* **Bộ nhớ:** 1536MB
* **Mã nguồn:** 50000 bytes

Phú ông nhận được thông tin về một trận mưa thiên thạch sắp ập xuống trái đất. Không những thế, Phú ông còn biết tọa độ của vị trí điểm rơi của mỗi một thiên thạch. Phú ông nhờ Cuội xác định xem có bao nhiêu thiên thạch có thể rơi xuống cánh đồng của ông ta. Cánh đồng của Phú ông có dạng một hình đa giác lồi được xác định bởi danh sách các đỉnh được liệt kê theo thứ tự ngược chiều kim đồng hồ.   
Yêu cầu: Xác định xem trong tập cho trước các điểm rơi của thiên thạch, có bao nhiêu điểm nằm trong cánh đồng của Phú ông. Các điểm nằm trên biên của cánh đồng không được tính là điểm nằm trong cánh đồng.

Input

- Dòng đầu tiên là số nguyên n (3 <= n <= 5000) là số đỉnh của đa giác lồi mô tả cánh đồng của Phú ông.   
- Mỗi dòng trong n dòng tiếp theo chứa cặp tọa độ của một đỉnh của đa giác lồi.   
- Dòng tiếp theo là số nguyên m (2 <= m <= 5000) - số thiên thạch rơi xuống.   
- Mỗi dòng trong số m dòng cuối cùng chứa 2 số là tọa độ điểm rơi của một thiên thạch.   
Các tọa độ là các số nguyên có trị tuyệt đối không quá 10^6.

Output

Ghi ra m dòng, mỗi dòng tương ứng với 1 điểm rơi của thiên thạch. Ghi "YES" nếu điểm rơi của thiên thạch nằm trong cánh đồng và ghi "NO" nếu trái lại.

Example

Input:

4

2 4

8 4

6 8

4 6

4

3 5

4 7

5 5

6 7

Output:

NO

NO

YES

YES

* **Người up:** dtmp
* **Điểm:** 0.24
* **Ngôn ngữ cho phép:**

Hệ thống đèn ( LIGHT )

**Giới hạn**

* **Thời gian:** 0.286s
* **Bộ nhớ:** 1536MB
* **Mã nguồn:** 50000 bytes

Khu vực đặt các bể xăng của một Tổng Công Ty Xăng Dầu có dạng một hình chữ nhật được chia thành m \* n ô vuông. Các ô vuông được đánh tọa độ 1 -> m từ trên xuống, 1 -> n từ trái sang.   
Tại k ô của lưới có đặt các bể xăng. Người ta cần xây dựng một hệ thống đèn pha chiếu sáng, mỗi đèn chỉ chiếu dọc theo hoặc là hàng hoặc là cột của lưới ô vuông sao cho mỗi bể chứa xăng phải được chiếu sáng bởi ít nhất một đèn pha chiếu dọc theo hàng hoặc cột chứa nó. Biết:   
- ai là chi phí xây dựng đèn chiếu sáng dọc theo hàng.   
- bj là chi phí xây dựng đèn chiếu sáng dọc theo cột.

Yêu cầu: Tìm cách xây dựng hệ thống đèn với tổng chi phí xây dựng là nhỏ nhất.

Input

- Dòng đầu tiên chứa 3 số nguyên dương m, n, k (m, n <= 100).   
- Dòng thứ hai chứa m số nguyên a1, a2, ..., am.   
- Dòng thứ ba chứa n số nguyên b1, b2, ..., bn.   
- Dòng thứ i trong k dòng tiếp theo chứa tọa độ của bể xăng thứ i.

Output

Một dòng duy nhất ghi tổng chi phí theo cách xây dựng tìm được.

Example

Input:

2 3 4

15 17

2 4 6

1 1

2 2

2 3

2 1

Output:

12

Đoạn con có tổng lớn nhất ( GSS )

**Giới hạn**

* **Thời gian:** 0.09s
* **Bộ nhớ:** 1536MB
* **Mã nguồn:** 50000 bytes

Cho dãy số a[1], a[2], ..., a[n] (|a[i]| <= 15000, n <= 50000).

Hàm q(x, y) = max { tổng(a[i]+a[i+1]+...+a[j]), x <= i <= j <= y }.

Cho m câu hỏi dạng x, y (1 <= x <= y <= n). (m <= 50000) -> hãy tính các q(x, y). 

Input

- Dòng đầu là n.

- Dòng thứ hai là dãy a.

- Dòng thứ 3 là m.

- m dòng tiếp theo mỗi dòng là 1 cặp số x, y.

Output

-> Lần lượt ghi ra các q(x, y) tương ứng. Mỗi kết quả ghi trên 1 dòng.

Example

**Input:**  
3  
-1 2 3  
1  
1 2  
**Output:**  
2

* **Người up:** dtmp
* **Điểm:** 0.15
* **Ngôn ngữ cho phép:**
* **Nguồn bài:** Bai` nay Hieu add day nhe ^^

Giá trị lớn nhất ( QMAX )

**Giới hạn**

* **Thời gian:** 0.689s
* **Bộ nhớ:** 1536MB
* **Mã nguồn:** 50000 bytes

Cho một dãy gồm n phần tử có giá trị ban đầu bằng 0.

Cho m phép biến đổi, mỗi phép có dạng (u, v, k): tăng mỗi phần tử từ vị trí u đến vị trí v lên k đơn vị.

Cho q câu hỏi, mỗi câu có dạng (u, v): cho biết phần tử có giá trị lớn nhất thuộc đoạn [u, v]

Giới hạn

* n, m, q <= 50000
* k > 0
* Giá trị của một phần tử luôn không vượt quá 2 31-1

Input

* Dòng 1: n, m
* m dòng tiếp theo, mỗi dòng chứa u, v, k cho biết một phép biến đổi
* Dòng thứ m+2: p
* p dòng tiếp theo, mỗi dòng chứa u, v cho biết một phép biến đổi

Output

* Gồm p dòng chứa kết quả tương ứng cho từng câu hỏi.

Example

Input:

6 2

1 3 2

4 6 3

1

3 4

Output:

3

* **Người up:** dtmp
* **Điểm:** 0.06
* **Ngôn ngữ cho phép:**

Bảo vệ ( BAOVE )

**Giới hạn**

* **Thời gian:** 0.419s
* **Bộ nhớ:** 1536MB
* **Mã nguồn:** 50000 bytes

Một mạng lưới gồm N thành phố, và một số đường một chiều nối các cặp thành phố (giữa hai thành phố có thể có nhiều đường nối một chiều).   
Quân địch đang tập trung ở thành phố N, định tiến công ta ở thành phố 1, và chúng sẽ tiến công trên tất cả các con đường chưa được bảo vệ để tiến vào thành phố 1. Bộ chỉ huy ta cần xác định số quân ít nhất trên các con đường để chặn địch tiến về thành phố 1.

Input

Dòng đầu ghi N (N ≤ 5000)   
Các dòng tiếp theo cho đến hết file, mỗi dòng một tả 1 đường gồm u, v, s cho biết có đoạn đường một chiều từ u đến v, và phải cần ít nhất s quân để chặn địch trên đường này. (s ≤ 65000)   
Có không quá 10000 đường.

Output

Số quân ít nhất cần điều động

Example

Input:

10

10 7 25050

6 1 12564

10 4 23916

5 1 61054

10 9 50950

9 1 35558

10 2 60941

3 1 22203

8 2 2853

5 7 31422

3 7 41491

8 7 27235

4 8 55965

8 6 41980

3 6 47707

2 3 45320

3 8 11237

7 6 38734

5 6 7561

3 5 8844

Output:

79169

* **Người up:** dtmp
* **Điểm:** 0.20
* **Ngôn ngữ cho phép:**

Sao đa giác ( POLY4 )

**Giới hạn**

* **Thời gian:** 0.116s
* **Bộ nhớ:** 1536MB
* **Mã nguồn:** 50000 bytes

Cho các đa giác không tự cắt, một đa giác được gọi là đa giác hình sao nếu như tồn tại một điểm nằm trong đa giác mà từ điểm đó có thể nhìn thấy tất cả các điểm nằm trong đa giác đó.   
Hai điểm là nhìn thấy nhau có nghĩa là khi nối chúng thì tạo thành 1 đoạn thẳng không cắt bất cứ một cạnh nào của đa giác.   
Hãy kiểm tra xem đa giác có phải là đa giác hình sao không?.

Input

Gồm một số test, mỗi test dòng đầu là n (n <= 50) (số đỉnh của đa giác).   
Tiếp theo là n cặp số nguyên x, y (-10000 <= x, y <= 10000) mô tả tọa độ của các đỉnh đi theo một chiều nhất định (ngược hoặc thuận chiều kim đồng hồ).   
Kết thúc test là số 0

Output

Ghi ra 1 hay 0 trên một dòng tương ứng nếu đa giác là hình sao hoặc không là hình sao.

Example

Input:

3

0 0

0 1

1 1

6

66 13

96 61

76 98

13 94

4 0

45 68

8

27 21

55 14

93 12

56 95

15 48

38 46

51 65

64 31

0

Output:

1

1

0

* **Người up:** dtmp
* **Điểm:** 0.73
* **Ngôn ngữ cho phép:**

Giá trị lớn nhất ver2 ( QMAX2 )

**Giới hạn**

* **Thời gian:** 0.421s
* **Bộ nhớ:** 1536MB
* **Mã nguồn:** 50000 bytes

Giống bài "Giá trị lớn nhất" ở trên.

Input

* n: số phần tử của dãy (n <= 50000).
* m: số lượng biến đổi và câu hỏi (m <= 100000).
  + biến đổi có dạng: 0 x y value
  + câu hỏi có dạng : 1 x y.

Output

Ghi ra trả lời cho lần lượt từng câu hỏi.

Example

Input:

6 3

0 1 3 3

0 4 6 4

1 1 6

Output:

4

* **Người up:** dtmp
* **Điểm:** 0.07
* **Ngôn ngữ cho phép:**

Bảng quan hệ ( REL7 )

**Giới hạn**

* **Thời gian:** 0.118s
* **Bộ nhớ:** 1536MB
* **Mã nguồn:** 50000 bytes

Cho một ma trận A kích thước n\*n chỉ gồm các giá trị { -1 , -2 , 0 , 1 , 2 , 3 }   
Giả thiết 2 <= n <= 300.   
Bảng A gọi là tương thích với dãy T = (t1, t2, ..., tn), hay dãy T tương thích với bảng A nếu:   
• Aij = 0 : ti = tj   
• Aij = 1 : ti < tj   
• Aij = -1 : ti > tj   
• Aij = 2 : ti <= tj   
• Aij = -2 : ti >= tj   
• Aij = 3 : ti khác tj   
(Với mọi i, j: 1 <= i, j <= n)   
Yêu cầu : cho trước bảng quan hệ A, hãy tìm dãy số nguyên dương T = (t1, t2, ..., tn) tương thích với bảng A mà max(T) là bé nhất có thể. Biết rằng luôn tồn tại một dãy như vậy .

Input

Dòng đầu tiên là số nguyên N .   
N dòng sau mỗi dòng gồm N số nguyên mô tả ma trận A.

Output

Dòng đầu tiên ghi ra max( T ) . Dòng thứ 2 ghi ra dãy số T1 , T2 , .. Tn . Mỗi số ghi cách nhau ít nhất một dấu cách .

Example

Input:

6

0 1 1 1 2 2

-2 0 1 0 2 2

-2 -1 0 3 0 1

-2 -2 3 0 1 1

-1 -2 0 -1 0 1

-1 -2 -1 -1 -1 0

Output:

4

1 2 3 2 3 4

* **Người up:** hard7771988
* **Điểm:** 1.95
* **Ngôn ngữ cho phép:**
* **Nguồn bài:** Mr Le Minh Hoang

Số lượng bậc ( DEGREE )

**Giới hạn**

* **Thời gian:** 0.065s
* **Bộ nhớ:** 1536MB
* **Mã nguồn:** 50000 bytes

Một số nguyên dương A gọi là có bậc K đối với cơ số B nếu như :   
• A = B^x1 + B^x2 + … + B^xk   
( trong đó x1 , x2 , … , xk là các số nguyên không âm thoả mãn x1 <> x2 <> x3 … <> xk )   
Ví dụ :   
• 17 có bậc 2 đối với cơ số 2 vì 17 = 2^4 + 2^0 .   
• 151 có bậc 3 đối với cơ số 5 vì 151 = 5^3 + 5^2 + 5^0.   
Yêu cầu : Cho trước 1 đoạn [X,Y] . Hãy xác định xem trong đoạn này có bao nhiêu số có bậc K đối với cơ số B.   
Giới hạn :   
• 1 <= X <= Y <= 10^9   
• 1 <= K <= 25, 2 <= B <= 9   
• Chạy được với bộ nhớ thông báo < 800 K bạn mới thực sự là thành công

Input

1 dòng gồm 4 số nguyên dương X , Y , K , B

Output

Gồm 1 dòng duy nhất ghi ra số lượng số tìm được .

Example

Input:

15 20 2 2

Output:

3

( Giải thích : Đó là các số 17 = 2^4 + 2^0 , 18 = 2^4 + 2^1 , 20 = 2^4 + 2^2 )

* **Người up:** hard7771988
* **Điểm:** 0.32
* **Ngôn ngữ cho phép:**
* **Nguồn bài:** Rybinsk State Avia Academy

Huyền thoại Lục Vân Tiên ( MINK )

**Giới hạn**

* **Thời gian:** 0.87s
* **Bộ nhớ:** 1536MB
* **Mã nguồn:** 50000 bytes

Dạo này tivi cũng đang chiếu phim Lục Vân Tiên , sẵn tiện lấy luôn làm tiêu đề .   
Lục Vân Tiên cũng giống Samurai Jack , bị Quan Thái Sư đẩy vào vòng xoáy thời gian và bị chuyển tới tương lai của những năm 2777 .   
Ở thời đại này , Tráng sỹ phải là người thông thạo máy tính , gõ bàn phím lia lịa như đấu sỹ thời xưa múa kiếm ấy và phải qua một cuộc thi lập trình mới được phong danh hiệu .   
Để vượt qua vòng loại , Vân Tiên cần tham gia cuộc thi sát hạch . Ban Giám Khảo cuộc thi sát hạch gồm có N người , họ đều là các cao thủ trong giới IT . Các thành viên trong Ban Giám Khảo được đánh số từ 1 -> N và mỗi người lại có một chỉ số sức mạnh gọi là APM ( Actions Per Minute ) . Các giám khảo sẽ xếp hàng lần lượt từ 1 -> N . Mỗi thí sinh sẽ phải đấu với K vị giám khảo và K vị giám khảo này phải đứng liền thành 1 đoạn ( Tức là i , i+1 , i+2 , ... i+K-1 ) , chỉ cần thắng 1 vị giám khảo thì sẽ vượt qua vòng loại .   
Tuy nhiên thí sinh kô được chọn xem những giám khảo nào sẽ đấu với mình .   
Vân Tiên rất lo vì lỡ may đụng độ với những vị giám khảo nào "khó nhằn" thì sẽ tiêu mất . Nên chiến thuật của Vân Tiên là tập trung hạ vị giám khảo có chỉ số APM thấp nhất trong số K vị . Bạn hãy lập trình để giúp Lục Vân Tiên xác định được ở tất cả các phương án thì chỉ số APM của vị giám khảo thấp nhất sẽ là bao nhiêu ( Có tất cả N-k+1 phương án :   
Phương án 1 : Vân Tiên phải đấu với vị 1 -> vị k   
Phương án 2 : Vân Tiên phải đấu với vị 2 -> vị k+1   
…   
Phương án N-k+1 : Vân Tiên phải đấu với vị N-k+1 -> vị N ) .   
  
( 1 <= N <= 17000 , chỉ số APM của 1 giám khảo >= 1 và <= 2 tỉ , 1 <= K <= N ) .

Bài này O(N) mới thực sự coi là accept . Còn lại O(NlogN) hay O(N\*K) thì bạn chỉ may mắn accept thôi .

Input

Dòng 1 : số T là số test .   
Tiếp theo là T bộ test , mỗi bộ test có format như sau :   
Dòng 1 : N k   
Dòng 2 : N số nguyên dương A[1] , … A[N] .

Output

Kết quả mỗi test ghi ra trên dòng , dòng thứ i gồm N-k+1 số , số thứ j tương ứng là chỉ số APM của vị giám khảo yếu nhất trong phương án j .

Example

Input:

2

4 2

3 2 4 1

3 3

1 2 3

Output:

2 2 1

1

* **Người up:** hard7771988
* **Điểm:** 0.06
* **Ngôn ngữ cho phép:**

Chiến trường Ô qua ( KAGAIN )

**Giới hạn**

* **Thời gian:** 0.2s
* **Bộ nhớ:** 1536MB
* **Mã nguồn:** 50000 bytes

Lại nói về Lục Vân Tiên , sau khi vượt qua vòng loại để trở thành Tráng Sỹ , anh đã gặp được Đôrêmon và được chú mèo máy cho đi quá giang về thế kỷ 19 . Trở lại quê hương sau nhiều năm xa cách , với tấm bằng Tráng Sỹ hạng 1 do Liên Đoàn Type Thuật cấp , anh đã được Đức Vua cử làm đại tướng thống lãnh 3 quân chống lại giặc Ô Qua xâm lăng . Đoàn quân của anh sẽ gồm N đại đội , đại đội i có A[i] ( > 0 ) người . Quân sỹ trong 1 đại đội sẽ đứng thành 1 cột từ người 1 -> người A[i] , như vậy binh sỹ sẽ đứng thành N cột . Vì Vân Tiên quyết 1 trận sẽ đánh bại quân Ô Qua nên đã cử ra 1 quân đoàn hùng mạnh nhất . Trong sử cũ chép rằng , quân đoàn của Vân Tiên cử ra lúc đó là một nhóm các đại đội có chỉ số liên tiếp nhau ( tức là đại đội i , i + 1 , … j ) . Vì sử sách thì mối mọt hết cả nên chỉ biết được mỗi thế . Ngoài ra theo giang hồ đồn đại thì sức mạnh của 1 quân đoàn = số người của đại đội ít người nhất \* số đại đội được chọn . Nhiệm vụ của bạn là dựa trên các thông số của các nhà khảo cổ có được , hãy cho biết quân đoàn mà Vân Tiên đã chọn ra là từ đại đội nào đến đại đội nào . Chú ý nếu có nhiều phương án thì ghi ra phương án mà chỉ số của đại đội đầu tiên được chọn là nhỏ nhất .

Bài này O(N) mới thực sự coi là accept . Còn lại O(NlogN) , O(N^2) thì đó là do bạn may mắn accept thôi.

Input

Dòng 1 : Số T là số bộ test .   
T nhóm dòng tiếp theo , mỗi nhóm dòng mô tả 1 bộ test . Nhóm dòng thứ i :   
Dòng 1: N ( <= 30000 )   
Dòng 2: N số nguyên mô tả N số A[1] , A[2] , … A[N] ( các số nguyên dương <= 30000 ).   
.

Output

Kết quả mỗi test ghi ra trên 1 dòng , gồm 3 số : sức mạnh quân đoàn mạnh nhất , chỉ số của đại đội đầu tiên và chỉ số của đại đội cuối cùng được chọn .

Example

Input:

2

4

3 4 3 1

4

1 2 1 3

Output:

9 1 3

4 1 4

* **Người up:** hard7771988
* **Điểm:** 0.08
* **Ngôn ngữ cho phép:**

Dạo chơi quanh nhà ( WALK )

**Giới hạn**

* **Thời gian:** 0.117s
* **Bộ nhớ:** 1536MB
* **Mã nguồn:** 10000 bytes

Cho N vectơ có độ dài không quá L . Xét đến vectơ i (x,y) ta có thể chọn vectơ i (x,y) hoặc vectơ –i ( -x,-y ) . Yêu cầu hãy xác định 1 cách chọn các vectơ sao cho vectơ tổng của N vectơ thì có độ dài không quá sqrt(2) \* L .   
1 <= N <= 10000 .   
1 <= L <= 1000 .

Input

Dòng 1 : 2 số nguyên N , L. N dòng tiếp theo mỗi dòng gồm 2 số nguyên xi , yi mô tả vectơ i .

Output

Gồm 1 dòng ghi ra N số nguyên , số thứ i là 1 tương ứng là chọn vectơ (xi,yi) và 2 tương ứng là chọn vectơ (-xi,-yi) . Dữ liêụ đảm bảo có phương án.

Example

Input:

4 5

5 0

0 5

0 0

-3 4

Output:

1 2 1 1

* **Người up:** hard7771988
* **Điểm:** 0.66
* **Ngôn ngữ cho phép:**
* **Nguồn bài:** VI Ural State University Collegiate Programming Contest (21.10.2001)

Vua Midas thích vàng ( MIDAS )

**Giới hạn**

* **Thời gian:** 0.044s
* **Bộ nhớ:** 1536MB
* **Mã nguồn:** 50000 bytes

Trong thần thoại Hy Lạp, do có công giúp đỡ người thầy của thần Dyonysus, nhà vua Midas được vị thần thưởng cho bằng cách đưa ngài đến một hòn đảo có dạng một đa giác lồi N cạnh. Hòn đảo đã được chia thành các vùng bởi M đường rào là các đoạn thẳng nối một số cặp đỉnh của đa giác. Biết rằng các đoạn rào chỉ có điểm chung tại các đầu mút (các đỉnh).   
Thần Dyonysus nói với Midas rằng, nhà vua có thể chọn K mảnh bất kì trong số các mảnh đã được tạo ra bởi đường biên của đảo và các đường rào, miễn sao K mảnh đó phải ghép lại thành một miền liên tục.   
Yêu cầu: Hãy giúp vua Midas tham lam chọn ra K mảnh sao cho miền thu được có diện tích lớn nhất.

Input

Dòng đầu tiên ghi 3 số nguyên dương N, M, K (K ≤ M ≤ 100, N ≤ 1000).   
N dòng tiếp theo, dòng thứ i ghi hai số nguyên xi, yi ( |xi|, |yi| <= 1000000) là tọa độ của đỉnh thứ i của đa giác lồi (theo chiều ngược kim đồng hồ).   
Cuối cùng là M dòng, mỗi dòng mô tả một đường rào, gồm 2 số u, v cho biết đường rào nối hai đỉnh u và v.

Output

1 Dòng duy nhất ghi S là diện tích lớn nhất tìm được, chính xác đến 2 chữ số sau dấu phẩy.

Example

**Input:**  
8 4 2  
0 0  
0 2  
1 4  
3 5  
5 5  
6 4  
6 2  
4 0  
1 3  
3 8  
4 7  
4 8  
  
**Output:**  
13.50

* **Người up:** hard7771988
* **Điểm:** 1.13
* **Ngôn ngữ cho phép:**
* **Nguồn bài:** Mr Tran Quang Khai

Câu chuyện người lính ( MILITARY )

**Giới hạn**

* **Thời gian:** 0.135s
* **Bộ nhớ:** 1536MB
* **Mã nguồn:** 50000 bytes

“Tôi vẫn nhớ chiến trường Điện Biên năm đó rất ác liệt, rất nhiều người lính đã ngã xuống. Tại vùng căn cứ này, địch cho xây dựng lô cốt, hàng rào dây thép gai rất nhiều , vòng trong nối vòng ngoài, tạo thành nhiều vòng bảo vệ … “Đó là dòng hồi tưởng của 1 người lính già đã từng tham gia chiến dịch Tây Bắc lịch sử. Lần theo những trang sử được ghi chép lại, người ta biết rằng tướng Đờ Cát lúc đầu chưa chọn vị trí để đặt sở chỉ huy mà tìm cách thiết lập các vòng bảo vệ bằng dây thép gai nối các cứ điểm lại với nhau, sau đó sẽ chọn đặt sở chỉ huy tại vị trí an toàn nhất là ở vị trí mà có nhiều vòng bảo vệ bao quanh nhất. Mỗi 1 vòng bảo vệ là 1 đa giác không tự cắt tạo thành bằng cách nối 1 số cứ điểm lại với nhau bằng dây thép gai, 1 cứ điểm thuộc về không quá 1 vòng bảo vệ, các vòng bảo vệ phải được thiết lập sao cho giữa 2 vòng bảo vệ bất kỳ X và Y thì phần diện tích chung của X và Y = Min( diện tích X, diện tích Y ) hoặc = 0. Trên mặt phẳng toạ độ, các cứ điểm được coi như các điểm có toạ độ nguyên. Bạn hãy xác định xem, sở chỉ huy của tướng Đờ Cát sẽ được bảo vệ tối đa bởi mấy vòng bảo vệ.   
  
Download một số test tại [đây](http://vn.spoj.pl/SPOJVN/content/MILITARY.rar)

Input

Dòng 1: số nguyên N là số cứ điểm. ( 1 ≤ N ≤ 4000 ).   
N dòng tiếp theo, dòng thứ i gồm 2 số nguyên xi, yi tương ứng là toạ độ của cứ điểm i . Các toạ độ đều là số nguyên dương ≤ 10000 .

Output

Gồm 1 dòng duy nhất ghi ra số lượng vòng bảo vệ tối đa mà sở chỉ huy của tướng Đờ Cát có thể được bao bọc .

Example

Input:

4

100 100

200 100

100 200

300 300

Output:

1

Giải thích: Ta nối cứ điểm 1, 2, 3, 4 lại tạo thành 1 vòng bảo vệ, đặt trụ sở chỉ huy bên trong thì ra được đáp án. Ngoài ra còn có các phương án khác là nối cứ điểm 1, 2, 3 tạo thành 1 vòng bảo vệ, nối cứ điểm 2, 3, 4 thành 1 vòng bảo vệ, … nhưng tất cả các phương án này thì khi chọn vị trí đặt trụ sở chỉ huy thì vẫn tối đa = 1.

* **Người up:** hard7771988
* **Điểm:** 0.40
* **Ngôn ngữ cho phép:**

Cái túi ( Hard version ) ( HUGEKNAP )

**Giới hạn**

* **Thời gian:** 0.1s
* **Bộ nhớ:** 1536MB
* **Mã nguồn:** 20000 bytes

Cho N đồ vật , vật i có khối lượng W[i] và giá trị là V[i] . Một cái túi có thể chịu được khối lượng tối đa là M , quá thì sẽ rách. Hãy tìm cách nhét 1 số đồ vật vào trong túi sao cho túi không bị rách và tổng giá trị của các đồ vật nhét vào là lớn nhất.   
  
Các tiêu chí yêu cầu : Chương trình của bạn chạy đúng , bộ nhớ thông báo ngoài status nhỏ hơn 850 KB ( với Free Pascal ) và nhỏ hơn 2850 KB ( với C++ ) thì bạn mới được coi là accept . Yêu cầu này đơn giản là vì bài này nguồn gốc là dành cho Turbo Pascal nên khi add lên trên này thì đành phải giới hạn với các bạn 1 chút .   
Nâng time limit từ 3s lên 5s cho các bạn dễ làm hơn vậy.

Input

Dòng đầu tiên là số nguyên T là số bộ test . ( 1 ≤ T ≤ 40 )   
Mỗi bộ test sẽ có format như sau :   
Dòng 1 : 2 số nguyên dương N , M ( 1 ≤ N ≤ 10000 , 1 ≤ M ≤ 1000 ) .   
Dòng 2 : Gồm N số nguyên là W[i] ( 1 ≤ W[i] ≤ 1000 ) .   
Dòng 3 : Gồm N số nguyên là V[i] ( 1 ≤ V[i] ≤ 10000 ) .

Output

Với mỗi bộ test :   
Dòng đầu tiên ghi ra giá trị lớn nhất có thể đạt được và số K là số đồ vật lựa chọn .   
Dòng thứ 2 ghi ra chỉ số của K đồ vật được chọn .

Example

Input:

1

3 4

1 2 3

4 5 6

Output:

10 2

1 3

* **Người up:** hard7771988
* **Điểm:** 0.30
* **Ngôn ngữ cho phép:**
* **Nguồn bài:** Folklore - Sách thầy Hoàng , mục bài tập tự giải , bài số 1 .

Floyd hoặc Dijkstra ( Cơ bản ) ( FLOYD )

**Giới hạn**

* **Thời gian:** 0.163s
* **Bộ nhớ:** 1536MB
* **Mã nguồn:** 50000 bytes

Cho đơn đồ thị vô hướng N đỉnh và M cạnh, trọng số các cạnh đều nguyên dương. Có 2 loại câu hỏi :   
0 u v : Cho biết đường đi ngắn nhất từ u tới v có độ dài là bao nhiêu.   
1 u v : Hãy chỉ ra 1 đường đi ngắn nhất từ u => v   
Bài cơ bản này nhằm kiểm tra kỹ năng xây dựng các module chương trình con dành cho truy vết 1 cách hợp lý, sử dụng nhuần nhuyễn chương trình con, lời gọi hàm .   
  
Download test và solution tại [đây](http://vn.spoj.pl/SPOJVN/content/floyd.rar)

Input

Dòng 1 : 3 số nguyên N , M , K . ( 1 ≤ N ≤ 100 , 1 ≤ M ≤ N\*(N-1)/2 , 1 ≤ K ≤ 1000 )   
M dòng tiếp theo , dòng thứ i gồm 3 số nguyên dương u , v , c cho biết cạnh (u,v) có trọng số là c ( 1 ≤ c ≤ 10000 )   
K dòng tiếp theo là K câu hỏi , dòng thứ j sẽ có định dạng như đã nêu ở trên .

Output

Ứng với mỗi câu hỏi trong K câu hỏi thì ta phải trả lời trên mỗi dòng như sau .   
Câu hỏi 0 u v : Ghi ra 1 số nguyên duy nhất là độ dài đường đi ngắn nhất từ u -> v.   
Câu hỏi 1 u v : Ghi ra số đầu tiên là số X là số đỉnh trên đường đi ngắn nhất này , tiếp đó ghi ra X số là chỉ số các đỉnh theo thứ tự xuất hiện trên hành trình .

Example

Input:

3 3 2

1 2 3

2 3 1

1 3 5

0 1 2

1 1 3

Output:

3

3 1 2 3

* **Người up:** hard7771988
* **Điểm:** 0.05
* **Ngôn ngữ cho phép:**

Một chút về Huffman Tree ( HEAP1 )

**Giới hạn**

* **Thời gian:** 0.659s
* **Bộ nhớ:** 1536MB
* **Mã nguồn:** 50000 bytes

Một người nông dân muốn cắt 1 thanh gỗ có độ dài L của mình thành N miếng , mỗi miếng có độ dài là 1 số nguyên dương A[i] ( A[1] + A[2] + … A[N] = L ) . Tuy nhiên để cắt một miếng gỗ có độ dài là X thành 2 phần thì ông ta sẽ mất X tiền . Ông nông dân này không giỏi tính toán lắm , vì vậy bạn được yêu cầu lập trình giúp ông ta cho biết cần để dành ít nhất bao nhiêu tiền thì mới có thể cắt được tấm gỗ như mong muốn .   
  
Lưu ý : Kết quả có thể vượt longint ( trong Pascal ) và vượt long ( trong C++ ) đấy nhé .

Input

Dòng 1 : 1 số nguyên dương T là số bộ test .   
T nhóm dòng tiếp theo mô tả các bộ test , mỗi nhóm dòng gồm 2 dòng :   
Dòng 1 : số nguyên dương N ( 1 ≤ N ≤ 20000 ) .   
Dòng 2 : N số nguyên dương A[1] ,…, A[N] . ( 1 ≤ A[i] ≤ 50000 )

Output

Kết quả mỗi test ghi ra trên 1 dòng , ghi ra 1 số nguyên dương duy nhất là chi phí tối thiểu cần để cắt tấm gỗ .

Example

Input:

1

4

1 2 3 4

Output:

19

Đầu tiên cắt miếng gỗ thành 2 phần có độ dài 6 và 4 . Sau đó cắt tiếp miếng có độ dài 6 -> 3 và 3 . Cắt 1 miếng 3 thành 2 phần có độ dài 1 , 2 . Như vậy chi phí là 10 + 6 + 3 = 19.

* **Người up:** hard7771988
* **Điểm:** 0.09
* **Ngôn ngữ cho phép:**
* **Nguồn bài:** USACO

Phép Xor ( XOR )

**Giới hạn**

* **Thời gian:** 0.065s
* **Bộ nhớ:** 1536MB
* **Mã nguồn:** 50000 bytes

Cho 1 tập N số nguyên dương A[1] , … A[N] . Tập số này được gọi là phụ thuộc tuyến tính nếu tồn tại 1 số nguyên A[i] nào đó thoả mãn :   
A[i] = A[j1] xor A[j2] … xor A[jk] ( với i , j1 , j2 , … , jk đôi một khác nhau và k là 1 số tuỳ ý ) . Nếu tập số này không phụ thuộc tuyến tính thì được gọi là độc lập tuyến tính .   
Hãy kiểm tra tập N số nguyên dương A[1] … A[N] có phải là độc lập tuyến tính hay không ? Nếu không hãy chỉ ra phải loại đi ít nhất bao nhiêu phần tử để tập còn lại là 1 tập độc lập tuyến tính .   
  
Download test tại [đây](http://vn.spoj.pl/SPOJVN/content/XOR.rar). Solution của bài này sẽ không được upload , các bạn phải tự giải.

Input

Dòng 1 : số nguyên dương T là số bộ test .   
Tiếp theo là T bộ test , mỗi bộ test có format như sau :   
Dòng 1 : số N ( 1 ≤ N ≤ 10000 ) .   
Dòng 2 : N số nguyên dương A[1] … A[N] .( 1 ≤ A[i] ≤ 2000000000 ) .

Output

Với mỗi test , nếu tập N số là độc lập tuyến tính thì ghi ra “YES” ngược lại ghi ra “NO X” với X là số số ít nhất cần phải bỏ đi để tập còn lại trở thành độc lập tuyến tính .

Example

Input:

2

2

1 2

3

1 2 3

Output:

YES

NO 1

* **Người up:** hard7771988
* **Điểm:** 1.10
* **Ngôn ngữ cho phép:**

Xây dựng lâu đài ( CASTLE )

**Giới hạn**

* **Thời gian:** 0.141s
* **Bộ nhớ:** 1536MB
* **Mã nguồn:** 20000 bytes

Vua Hùng Vương thứ 18 đang muốn xây dựng cho mình một lâu đài trên một khu đất ở Phú Thọ. Để cho đơn giản, ta sẽ tọa độ hóa khu đất theo hệ tọa độ Đề Các. Theo các kiến trúc sư, để đảm bảo các yêu cầu mỹ quan, lâu đài nên được xây theo hình chữ nhật có các cạnh song song trục toạ độ. Còn theo thuật phong thủy, các góc của lâu đài không thể xây đặt tuỳ ý mà chỉ có thể đặt ở một số vị trí. Nhà vua muốn xây lâu đài cho mình vừa đẹp lại vừa hợp phong thuỷ. Và bạn, một chuyên gia lập trình, được yêu cầu giúp đỡ nhà vua. Nhà vua còn muốn biết có bao nhiêu cách xây lâu đài, và trong đó có bao nhiêu cách xây có diện tích lớn nhất và có diện tích nhỏ nhất, các diện tích đó là bao nhiêu?

Lưu ý các bạn cài bài này phải cài thật tinh tế. Solution mẫu của tác giả bài này mình đã submit thử để đảm bảo nó bị chạy quá thời gian (Time Limit Exceeded, kích vào link xem All Submissions ở trên) . O(N^2) là thuật toán chuẩn, O(N^2\*logN) nếu cài tốt thì cũng có thể accept.

Input

Dòng thứ nhất ghi số N là số điểm có thể đặt góc lâu đài.( 1 ≤ N ≤ 5000 )  
N dòng sau ghi 2\*N số nguyên là tọa độ N điểm có thể đặt góc lâu đài. ( Tọa độ của các điểm là các số nguyên có giá trị tuyệt đối nhỏ hơn 2^31 )

Output

Dòng thứ nhất ghi số M là số cách xây lâu đài.  
Nếu M > 0, dòng thứ hai ghi 2 số SMAX, CMAX là diện tích lớn nhất của lâu đài và số lượng lâu đài có diện tích là SMAX.  
Dòng thứ 3 ghi 2 số SMIN, CMIN là diện tích nhỏ nhất của lâu đài và số lượng lâu đài có diện tích là SMIN.

Example

**Input:**

8

1 1

1 2

1 3

2 1

2 2

2 3

3 1

3 2

**Output:**

5

2 2

1 3

* **Người up:** hard7771988
* **Điểm:** 0.52
* **Ngôn ngữ cho phép:**
* **Nguồn bài:** AGAMES 2004

Lập lịch sửa chữa ô tô ( CAR )

**Giới hạn**

* **Thời gian:** 0.185s
* **Bộ nhớ:** 1536MB
* **Mã nguồn:** 20000 bytes

Một cơ sở sửa chữa ô tô có nhận n chiếc xe để sửa. Do các nhân viên làm việc quá lười nhác nên đã đến hạn trả cho khách hàng mà vẫn chưa tiến hành sửa được chiếc xe nào. Theo hợp đồng đã ký kết từ trước, nếu bàn giao xe thứ i quá hạn ngày nào thì sẽ phải trả thêm một khoản tiền phạt là A[i].   
Ông chủ cơ sở sửa chữa quyết định sa thải toàn bộ công nhân và thuê nhân công mới. Với lực lượng mới này, ông ta dự định rằng để sửa chiếc xe thứ i sẽ cần B[i] ngày. Vấn đề đặt ra đối với ông là phải lập lịch sửa tuần tự các chiếc xe sao cho tổng số tiền bị phạt là ít nhất.   
  
Yêu cầu: Hãy lập lịch sửa xe giúp cho ông chủ cơ sở sửa chữa ô tô.   
  
Download test và solution tại [đây](https://www.spoj.pl/CSP/content/CAR.rar).

Input

• Dòng 1: Chứa số n (n ≤ 10000)   
• Dòng 2: Chứa n số nguyên dương A[1], A[2], ..., A[n] (1 ≤ A[i] ≤ 10000)   
• Dòng 3: Chứa n số nguyên dương B[1], B[2], ..., B[n] (1 ≤ B[i] ≤ 100)

Output

• Dòng 1: Ghi số tiền bị phạt tối thiểu   
• Dòng 2: Ghi số hiệu các xe sẽ tiến hành sửa chữa, theo thứ tự từ xe được sửa đầu tiên đến xe sửa sau cùng

Example

Input:

4

1 3 4 2

3 2 3 1

Output:

44

4 2 3 1

Xong công việc 4 vào cuối ngày 1 => phải trả 2 \* 1 = 2 .   
Xong công việc 2 vào cuối ngày 3 => phải trả 3 \* 3 = 9.   
Xong công việc 3 vào cuối ngày 6 => phải trả 6 \* 4 = 24 .   
Xong công việc 1 vào cuối ngày 9 => phải trả 1 \* 9 = 9 .   
Vậy tổng cộng phải trả 44 .

* **Người up:** hard7771988
* **Điểm:** 0.09
* **Ngôn ngữ cho phép:**
* **Nguồn bài:** Mr Le Minh Hoang

ập lịch trên 2 máy ( TWO )

**Giới hạn**

* **Thời gian:** 0.259s
* **Bộ nhớ:** 1536MB
* **Mã nguồn:** 20000 bytes

Có N chi tiết máy cần được gia công lần lượt trên 2 máy A và B. Thời gian gia công chi tiết i trên máy A là a[i], thời gian gia công trên máy B là b[i]. Hãy tìm trình tự gia công các chi tiết trên 2 máy sao cho việc hoàn thành gia công tất cả các chi tiết là sớm nhất có thể.

Input

* Dòng 1: số nguyên dương N (1 ≤ N ≤ 10000).
* Dòng 2: N số nguyên dương a[1], …, a[n]. (1 ≤ a[i] ≤ 10000)
* Dòng 3: N số nguyên dương b[1], …, b[n]. (1 ≤ b[i] ≤ 10000)

Output

* Dòng 1: Số nguyên dương T là thời điểm sớm nhất có thể hoàn thành.
* Dòng 2: N số nguyên là lịch trình gia công các chi tiết máy.

Example

Input:

3

2 3 1

1 2 3

Output:

7

3 2 1

* **Người up:** hard7771988
* **Điểm:** 0.13
* **Ngôn ngữ cho phép:**
* **Nguồn bài:** Folklore

Lập lịch trên 3 máy ( MACHINE )

**Giới hạn**

* **Thời gian:** 0.134s
* **Bộ nhớ:** 1536MB
* **Mã nguồn:** 20000 bytes

Có N chi tiết máy cần được gia công lần lượt trên 3 máy A , B và C. Thời gian gia công chi tiết i trên máy A là a[i] , thời gian gia công trên máy B là b[i] , thời gian gia công trên máy C là c[i] . Biết rằng 1 trong 2 điều kiện sau đây được thoả mãn : max( b[i] ) ≤ min( a[i] ) hoặc max( b[i] ) ≤ min( c[i] ) ( i = 1,…n ) .   
Hãy tìm trình tự gia công các chi tiết trên 3 máy sao cho việc hoàn thành gia công tất cả các chi tiết là sớm nhất có thể .

Input

Dòng 1 : số nguyên dương N ( 1 ≤ N ≤ 10000 ) .   
Dòng 2 : N số nguyên dương a[1] , … a[n] . ( 1 ≤ a[i] ≤ 10000 )   
Dòng 3 : N số nguyên dương b[1] , … b[n] .( 1 ≤ b[i] ≤ 10000 )   
Dòng 4 : N số nguyên dương c[1] , … c[n] .( 1 ≤ c[i] ≤ 10000 )

Output

Dòng 1 : Số nguyên dương T là thời điểm sớm nhất có thể hoàn thành .   
Dòng 2 : N số nguyên là lịch trình gia công các chi tiết máy .

Example

Input:

2

1 2

3 2

4 4

Output:

12

1 2

* **Người up:** hard7771988
* **Điểm:** 0.32
* **Ngôn ngữ cho phép:**
* **Nguồn bài:** Folklore

**TRANG 2**

Dãy nghịch thế độ dài K ( KINV )

**Giới hạn**

* **Thời gian:** 0.088s
* **Bộ nhớ:** 1536MB
* **Mã nguồn:** 20000 bytes

Cho dãy N số nguyên dương A[1] , … A[N] là một hoán vị của 1 , 2 , 3 , … N .   
Một dãy nghịch thế độ dài k là 1 dãy A[j1] > A[j2] > A[j3] … > A[jk] với j1 < j2 < j3 … < jk . Hãy đếm xem có tất cả bao nhiêu dãy nghịch thế độ dài k .

Input

Dòng 1 : 2 số nguyên dương N và k ( 2 ≤ N ≤ 10000 , 2 ≤ k ≤ 10 ) .   
Dòng 2 : N số nguyên dương A[1] … A[N] .

Output

Giả sử T là số lượng dãy nghịch thế có độ dài k , hãy ghi ra T mod 10^9 .

Example

Input:

3 2

3 2 1

Output:

3

* **Người up:** hard7771988
* **Điểm:** 0.16
* **Ngôn ngữ cho phép:**
* **Nguồn bài:** Dmitry Gozman Contest 1

Đảo giấu vàng ( GOLD )

**Giới hạn**

* **Thời gian:** 0.095s
* **Bộ nhớ:** 1536MB
* **Mã nguồn:** 50000 bytes

1 nhà thám hiểm nọ vừa phát hiện ra một bản đồ kho báu . Trên bản đồ miêu tả 1 hòn đảo nằm ở nam Thái Bình Dương . Trên hòn đảo có N vị trí có kho báu là các mỏ vàng . Để được phép khai thác nhà thám hiểm quyết định dốc hết tiền của ra mua 1 mảnh đất và khai thác các mỏ vàng trên đó . Tuy nhiên Nhà thám hiểm cũng không giàu có gì lắm nên chỉ có thể mua được 1 miếng đất hình chữ nhật có kích thước tối đa là S \* W và theo yêu cầu của Chúa Đảo thì miếng đất phải song song với 2 trục Ox và Oy để không làm mất mỹ quan của hòn đảo ( các mỏ vàng nằm trên đường biên của miếng đất cũng sẽ được quyền khai thác ) . Bạn hãy lập trình giúp Nhà thám hiểm tính xem ông ta có thể chiếm được nhiều nhất là bao nhiêu mỏ vàng .   
Lưu ý : Bài này nếu không cẩn thận sẽ rất dễ bị ngộ nhận . Vì vậy nên phải đặc biệt chú ý. Là 1 bài khó vì thế nên sau một thời gian sẽ để cho các bạn có thể xem lời giải và download test .   
  
Download test và solution tại [đây](http://vn.spoj.pl/SPOJVN/content/GOLD.rar).

Input

Dòng 1 : 2 số nguyên dương S W ( 1 ≤ S , W ≤ 10000 ). S là độ dài cạnh song song với trục Ox . W là độ dài cạnh song song với trục Oy .   
Dòng 2 : số nguyên dương N ( 1 ≤ N ≤ 15000 ) .   
N dòng tiếp theo , dòng thứ i mô tả vị trí của mỏ vàng thứ i là 2 số nguyên xi và yi . ( -30000 ≤ xi , yi ≤ 30000 ) .

Output

Gồm 1 dòng duy nhất ghi ra số lượng nhiều nhất mỏ vàng mà Nhà thám hiểm có thể có được .

Example

Input:

1 2

12

0 0

1 1

2 2

3 3

4 5

5 5

4 2

1 4

0 5

5 0

2 3

3 2

Output:

4

* **Người up:** hard7771988
* **Điểm:** 0.32
* **Ngôn ngữ cho phép:**
* **Nguồn bài:** Polish Olympiad in Informatics

Cây P đỉnh ( Cơ bản ) ( PTREE )

**Giới hạn**

* **Thời gian:** 0.152s
* **Bộ nhớ:** 1536MB
* **Mã nguồn:** 20000 bytes

Cho một cây gồm N đỉnh , mỗi đỉnh có 1 nhãn C[i] gọi là trọng số của đỉnh i . Hãy tìm 1 cây con gồm P đỉnh sao cho tổng trọng số của cây con này là lớn nhất . Hiểu 1 cách đơn giản là tìm P đỉnh sao cho P đỉnh này liên thông và tổng trọng số là lớn nhất .   
  
Vì bài này là ở mức độ khó tuy nhiên lại có nhiều ứng dụng nên các bạn có thể xem lời giải, download test. Nói chung cũng có thể xếp bài này vào nhóm bài “cơ bản” .   
  
Download test và solution tại [đây](http://vn.spoj.pl/SPOJVN/content/ptree.rar).

Input

Dòng 1 : 2 số nguyên dương N và P . ( 1 ≤ P ≤ N ≤ 200 ) .   
Dòng 2 : N số nguyên dương C[1] , … C[N] . ( -1000 ≤ C[i] ≤ 1000 ) .   
N – 1 dòng tiếp theo , mỗi dòng gồm 2 số nguyên dương u , v mô tả 1 cạnh của đồ thị .

Output

Gồm 1 dòng ghi ra P số nguyên là chỉ số của P đỉnh được chọn .

Example

Input:

3 2

1 2 3

1 2

2 3

Output:

2 3

* **Người up:** hard7771988
* **Điểm:** 0.19
* **Ngôn ngữ cho phép:**
* **Nguồn bài:** Folklore

Mạng điện ( ENET )

**Giới hạn**

* **Thời gian:** 0.033s
* **Bộ nhớ:** 1536MB
* **Mã nguồn:** 20000 bytes

Xét một mạng điện gồm N nút (đánh số từ 1 đến N) và hệ thống gồm M đường dây , mỗi đường dây nối trực tiếp một cặp nút nào đó của mạng . Với mục đính khảo sát hiệu thế giữa hai nút s, t nào đó của mạng ảnh hưởng đến điện áp của các nút trong mạng, người ta muốn xác định các nút gọi là các nút thế năng của mạng. Một nút của mạng được gọi là nút thế năng nếu như việc truyền tải điện năng từ nút s đến nút t trên mạng có thể thực hiện theo tuyến đường dây có đi qua nút này đồng thời mỗi nút của mạng xuất hiện trên tuyến đường dây này không quá một lần.   
Yêu cầu: Xác định tất cả các nút thế năng của mạng điện .   
  
Download test tại [đây](http://vn.spoj.pl/SPOJVN/content/ENET.rar). Solution của bài này sẽ không được upload , các bạn phải tự giải. Lưu ý là đây là bộ test thử , còn bộ test dùng trong chương trình check có thể khác .

Input

Dòng đầu tiên chứa bốn số N, M, s, t (N ≤ 1000, M ≤ 15000).   
Dòng thứ i trong M dòng tiếp theo chứa hai số Di, Ci là các số hiệu hai nút tương ứng hai đầu mút của đường dây thứ i.

Output

Dòng đầu tiên ghi số K là số lượng nút thế năng tìm được.   
Dòng thứ i trong K dòng cuối cùng ghi số hiệu của nút thế năng thứ i , các chỉ số được ghi theo thứ tự tăng dần .

Example

Input:

3 2 1 3

3 1

1 2

Output:

2

1

3

* **Người up:** hard7771988
* **Điểm:** 0.39
* **Ngôn ngữ cho phép:**
* **Nguồn bài:** Mr Tran Quang Khai

Diện tích hình chữ nhật ( AREA )

**Giới hạn**

* **Thời gian:** 0.075s
* **Bộ nhớ:** 1536MB
* **Mã nguồn:** 50000 bytes

Trên mặt phẳng toạ độ người ta vẽ ra N hình chữ nhật . Hãy tính diện tích che phủ bởi N hình chữ nhật này , biết rằng N hình chữ nhật này song song với 2 trục Ox và Oy .

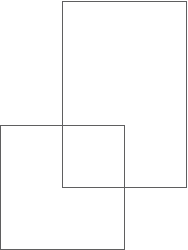
Input

Dòng 1 : số nguyên N ( 1 ≤ N ≤ 10000 ) .   
N dòng tiếp theo , mỗi dòng gồm 4 số nguyên x1 , y1 , x2 , y2 tương ứng là toạ độ góc trái dưới và góc phải trên của hình chữ nhật thứ i.( 0 ≤ x1 ≤ x2 ≤ 30000 , 0 ≤ y1≤ y2 ≤ 30000 ) .

Output

Gồm 1 dòng ghi ra diện tích phủ bởi N hình chữ nhật

Example



Input:

2

10 10 20 20

15 15 25 30

Output:

225

* **Người up:** hard7771988
* **Điểm:** 0.19
* **Ngôn ngữ cho phép:**
* **Nguồn bài:** BOI 2001

Thả trứng , trò giải trí tuổi teen ( EGG )

**Giới hạn**

* **Thời gian:** 0.2s
* **Bộ nhớ:** 1536MB
* **Mã nguồn:** 50000 bytes

Tụi trẻ con có được N quả trứng có độ cứng như nhau . Trong giờ ra chơi chúng quyết định thử xem trứng cứng đến mức nào bằng cách thả trứng từ trên tầng cao xuống đất xem ở độ cao nào thì trứng sẽ vỡ . Giả sử độ cứng của trứng là E thì thả trứng ở các tầng từ tầng 1 -> tầng E trứng sẽ không vỡ , và thả trứng bắt đầu ở tầng E + 1 trở đi trứng sẽ vỡ . Cách làm nông dân nhất là ta cứ đem thả từng tầng một từ thấp lên cao đến tầng nào trứng vỡ là biết ngay nhưng mà như thế phải thả nhiều lần quá , giờ ra chơi của bọn trẻ con không được lâu đến thế , hơn nữa tụi nó có tới N quả trứng nên có vỡ 1 số quả cũng chẳng sao , miễn là đạt được mục đích của mình .   
  
Bạn là một lập trình viên siêu hạng , sau khi nghe nỗi niềm của bọn trẻ , bạn có thể giúp gì được bọn trẻ không ? Hay là sẽ chịu thua ? Nếu giải được bài toán hóc búa này thì bạn hãy thử submit xem nào . Biết rằng toà nhà trường học của bọn trẻ có tất cả M tầng ( nếu trứng không vỡ ở tầng M thì có thể coi như nó có độ cứng là M ) .   
  
Chú ý nếu không cẩn thận sẽ rất dễ bị ngộ nhận . Dù làm cách nào đi nữa thì vấn đề muôn thuở vẫn là phải chứng minh được tính đúng đắn của thuật toán .

Input

Dòng 1 : số test T ( 1 ≤ T ≤ 10000 ) .   
T dòng tiếp theo mỗi dòng gồm 2 số nguyên N M ( 1 ≤ N , M ≤ 1000 ) .

Output

Với mỗi test ghi ra số lượng lần thả ít nhất ( X ) để có thể xác định được rõ ràng độ cứng của quả trứng ( kể cả trong trường hợp xấu nhất thì với X lần thả cũng có thể xác định được độ cứng của quả trứng ) .

Example

Input:

2

1 10

2 5

Output:

10

3

Giải thích test 1 ( N=1, M=10 ) : Giả sử ta bắt đầu thả trứng ở tầng 5 . Nếu trứng vỡ -> ta không còn trứng để thử nữa ( vì ta có mỗi một quả trứng ) -> không thể xác định được độ cứng của trứng là 0 hay 1, 2, 3, 4 . Nếu ta thả trứng từ tầng 1 , trứng vỡ -> độ cứng của trứng là 0 , nếu không vỡ ta lại thả tiếp ở tầng thứ 2 , … cứ làm như vậy thì ở trường hợp tệ nhất là trứng có độ cứng là 10 thì ta phải mất tới 10 lần thử .

* **Người up:** hard7771988
* **Điểm:** 0.24
* **Ngôn ngữ cho phép:**
* **Nguồn bài:** Folklore

Cặp ghép không trọng số ( MATCH1 )

**Giới hạn**

* **Thời gian:** 0.209s
* **Bộ nhớ:** 1536MB
* **Mã nguồn:** 50000 bytes

Cho đồ thị hai phía G = (X U Y, E); Các đỉnh của X ký hiệu là x1, x2, ..., xm, các đỉnh của Y ký hiệu là y1, y2, ..., yn.   
Một bộ ghép trên G là một tập các cạnh thuộc E đôi một không có đỉnh chung.   
  
Yêu cầu: Hãy tìm bộ ghép cực đại (có nhiều cạnh nhất) trên G.   
  
Chú ý : Dùng Eof chứ không dùng SeekEof.

Input

• Dòng 1: Chứa hai số m, n (1 ≤ m, n ≤ 100)   
• Các dòng tiếp, mỗi dòng chứa hai số nguyên dương i, j cho biết thông tin về một cạnh (xi, yj) thuộc E.

Output

• Dòng 1: Ghi số cạnh trong bộ ghép cực đại tìm được (K).   
• K dòng tiếp theo, mỗi dòng ghi thông tin về một cạnh được chọn vào bộ ghép cực đại: Gồm 2 số u, v thể hiện cho cạnh nối (xu, yv).

Example

Input:

4 5

1 1

1 4

2 1

2 2

2 4

3 2

3 3

4 2

4 3

Output:

4

1 1

2 4

3 3

4 2

* **Người up:** hard7771988
* **Điểm:** 0.11
* **Ngôn ngữ cho phép:**
* **Nguồn bài:** Cuốn DSAP của thầy Lê Minh Hoàng, giáo viên khối chuyên Sư Phạm

Bộ ghép đầy đủ trọng số cực tiểu ( MATCH2 )

**Giới hạn**

* **Thời gian:** 0.118s
* **Bộ nhớ:** 1536MB
* **Mã nguồn:** 50000 bytes

Cho đồ thị hai phía G = (X U Y, E); Các đỉnh của X ký hiệu là x1, x2, ..., xn, các đỉnh của Y ký hiệu là y1, y2, ..., yn. Mỗi cạnh của G được gán một trọng số không âm. Một bộ ghép đầy đủ trên G là một tập n cạnh thuộc E đôi một không có đỉnh chung. Trọng số của bộ ghép là tổng trọng số các cạnh nằm trong bộ ghép.   
  
Ràng buộc: Luôn tồn tại ít nhất một bộ ghép đầy đủ trên G.   
Chú ý dùng Eof chứ không dùng SeekEof

Input

• Dòng 1: Chứa số n (1 ≤ n ≤ 200)   
• Các dòng tiếp theo, mỗi dòng chứa 3 số nguyên i, j, c cho biết có một cạnh (xi, yj) và trọng số cạnh đó là c (0 ≤ c ≤ 200).

Output

• Dòng 1: Ghi trọng số bộ ghép tìm được   
• n dòng tiếp, mỗi dòng ghi hai số (u, v) tượng trưng cho một cạnh (xu, yv) được chọn vào bộ ghép.

Example

Input:

4

1 1 0

1 2 0

2 1 0

2 4 2

3 2 1

3 3 0

4 3 0

4 4 9

Output:

3

1 1

2 4

3 2

4 3

* **Người up:** hard7771988
* **Điểm:** 0.20
* **Ngôn ngữ cho phép:**
* **Nguồn bài:** Sách DSAP của thầy Lê Minh Hoàng , giáo viên khối chuyên Sư Phạm

Giao lưu ( FLOW1 )

**Giới hạn**

* **Thời gian:** 0.05s
* **Bộ nhớ:** 1536MB
* **Mã nguồn:** 50000 bytes

Cuộc thi giao lưu "Tết Ta Tin (TTT)" giữa hai đội Sư Phạm (SP) và Tổng Hợp (TH) có m bài toán tin học, mỗi đội có n học sinh tham dự. Các bài toán được đánh số từ 1 đến m và các học sinh của mỗi đội được đánh số từ 1 tới n.   
Học sinh của hai đội đều là những lập trình viên xuất sắc, tuy nhiên mỗi học sinh có thể giải quyết những bài toán thuộc sở trường của mình hiệu quả hơn những bài khác.   
  
Hãy giúp thầy My tổ chức cuộc thi theo thể thức sau:   
• Chọn đúng n cặp đấu, mỗi cặp gồm 01 học sinh SP và 01 học sinh TH làm 01 bài toán trong số những bài toán này.   
• Có đúng n bài toán được mang ra thi   
• Học sinh nào cũng được tham gia   
• Bài toán cho cặp đấu bất kỳ phải thuộc sở trường của cả hai thí sinh trong cặp   
  
Biết rằng luôn tồn tại phương án thực hiện yêu cầu trên.

Input

• Dòng 1: Chứa hai số n, m (1 ≤ n ≤ m ≤ 255)   
• n dòng tiếp theo, dòng thứ i ghi danh sách các bài toán thuộc sở trường của học sinh SP thứ i.   
• n dòng tiếp theo, dòng thứ j ghi danh sách các bài toán thuộc sở trường của học sinh TH thứ j.   
Chú ý dùng Eoln chứ không dùng SeekEoln

Output

Gồm m dòng, dòng thứ k ghi số hiệu thí sinh SP và số hiệu thí sinh TH trong cặp đấu bằng bài toán k, nếu bài toán k không được mang ra thi thì ghi vào dòng này hai số 0 .

Example

Input:

4 6

3 6

1 2

2 4

5

6

3 5 6

4

1 2 6

Output:

2 4

0 0

0 0

3 3

4 2

1 1

* **Người up:** hard7771988
* **Điểm:** 0.38
* **Ngôn ngữ cho phép:**
* **Nguồn bài:** Tuyển tập 150 bài của thầy Lê Minh Hoàng

Tô màu nhỏ nhất ( CTREE )

**Giới hạn**

* **Thời gian:** 0.098s
* **Bộ nhớ:** 1536MB
* **Mã nguồn:** 20000 bytes

Cho một cây gồm N nút, hãy tìm cách gán mỗi đỉnh một nhãn nguyên dương sao cho:   
+ Hai nút có cạnh nối được gán bởi hai nhãn khác nhau.   
+ Tổng giá trị các nhãn là nhỏ nhất.

Input

Dòng đầu tiên ghi N ( 1 ≤ N ≤ 10000).   
N-1 dòng tiếp theo, mỗi dòng ghi hai nút là hai đầu mút của một cạnh thuộc cây.

Output

Dòng đầu tiên ghi S là tổng giá trị nhãn tìm được.   
N dòng tiếp theo, dòng thứ i ghi nhãn gán cho đỉnh i trong phép gán tối ưu tìm được.

Example

Input:

8

1 2

1 3

1 4

1 5

5 6

5 7

5 8

Output:

11

3

1

1

1

2

1

1

1

* **Người up:** hard7771988
* **Điểm:** 0.17
* **Ngôn ngữ cho phép:**
* **Nguồn bài:** Mr Tran Quang Khai

Du lịch ( TWOSAT )

**Giới hạn**

* **Thời gian:** 0.066s
* **Bộ nhớ:** 1536MB
* **Mã nguồn:** 30000 bytes

Một công ty du lịch tổ chức cho 1 đoàn du khách nước ngoài đi du lịch M thành phố ở Việt Nam . Tuy nhiên mỗi du khách lại có 2 yêu cầu . Mỗi yêu cầu có dạng “Không muốn đi thành phố A” hoặc “Muốn đi thành phố A” ( A là chỉ số thành phố mà người đó yêu cầu ) . ( Có thể có trường hợp 2 yêu cầu của khách là “Muốn đi thành phố A” và Vì các du khách này là người nước ngoài nên rất khó tính , họ muốn ít nhất 1 trong 2 yêu cầu của họ phải được đáp ứng . Bên công ty du lịch đã đau đầu tìm cách chọn ra các thành phố để đưa đoàn du khách đi mà vẫn chưa tìm được cách nào cả . Bạn được yêu cầu giúp công ty du lịch này chọn ra 1 số thành phố để đưa đoàn du khách này đi mà lại vừa thoả mãn được các du khách này .

Input

Dòng 1 : 2 số nguyên N và M ( 1 ≤ N ≤ 20000 , 1 ≤ M ≤ 8000 ) tương ứng là số khách du lịch và số thành phố .   
M dòng tiếp theo gồm 2 số nguyên u , v , -M ≤ u ,v ≤ M ( u <> 0 , v <> 0 ) mô tả yêu cầu của khách thứ i ( số dương nếu yêu cầu du khách i muốn đi thành phố đó và số âm nếu không muốn đi thành phố đó ).

Output

Dòng 1 : Ghi YES nếu có phương án thoả mãn yêu cầu các du khách và ghi NO trong trường hợp ngược lại .   
Nếu YES thì ghi tiếp theo như sau :   
Dòng 2 : số nguyên dương K là số thành phố được chọn .   
Dòng 3 : Gồm K số nguyên là chỉ số của các thành phố được chọn .

Example

Input:

2 3

-1 -2

1 2

Output:

YES

2

2 3

* **Người up:** hard7771988
* **Điểm:** 0.58
* **Ngôn ngữ cho phép:**
* **Nguồn bài:** Dựa theo BOI 2001

Dịch chuyển tức thời ( TELEPORT )

**Giới hạn**

* **Thời gian:** 0.646s
* **Bộ nhớ:** 1536MB
* **Mã nguồn:** 20000 bytes

Pháp sư vĩ đại Byter đã phù phép tạo nên 2 hòn đảo trên biển Baltic : đảoBornholm và đảo Gotland . Ở mỗi đảo thì ông cũng tạo nên một vài cổng dịch chuyển tức thời ( CDCTT ) . Mỗi CDCTT sẽ là 1 trong 2 loại hình sau :   
1 ) Cổng Đến : Người ta sẽ được di chuyển tới cổng này .   
2 ) Cổng Đi : Khi bước vào cổng này người ta sẽ được đưa tới 1 Cổng Đến duy nhất xác định nằm ở hòn đảo kia .   
  
Một lần Byter đã giao cho các học trò của mình bài toán như sau : Cho biết số lượng CDCTT ở mỗi hòn đảo . Các học trò phải xác định xem cổng nào sẽ là Cổng Đến , cổng nào sẽ là Cổng Đi sao cho thoả mãn yêu cầu sau : Giả sử cổng i được đặt là Cổng Đến thì có ít nhất 1 Cổng Đi sẽ đưa người được dịch chuyển tới cổng i này và ngược lại , cổng i được đặt là Cổng Đi thì cổng mà nó gửi người đến phải được đặt là Cổng Đến .

Input

Dòng 1 : 2 số nguyên N và M ( 1 ≤ N , M ≤ 50000 ) tương ứng là số CDCTT ở trên đảo Bornhom và Gotland .   
Dòng thứ 2 gồm N số nguyên A[1] … A[N] mô tả các CDCTT ở đảo Bornhom : số A[i] cho biết nếu như Cổng thứ i trên đảo Bornhom được đặt là Cổng Đi thì nó sẽ gửi người đến Cổng A[i] trên đảo Gotland . ( 1 ≤ A[i] ≤ M ) .   
Dòng thứ 3 gồm M số nguyên B[1] … B[M] mô tả các CDCTT ở đảo Gotland : số B[i] cho biết nếu như Cổng thứ i trên đảo Gotland được đặt là Cổng Đi thì nó sẽ gửi người đến Cổng B[i] trên đảo Bornhom . ( 1 ≤ B[i] ≤ N ) .

Output

Dòng 1 : N số nguyên C[1] … C[N] ghi cách nhau 1 dấu khoảng trắng , C[i] = 1 nếu cổng i trên đảo Bornhom là Cổng Đi và = 0 nếu cổng i là Cổng Đến .   
Dòng 2 : M số nguyên D[1] … D[M] ghi cách nhau 1 khoảng trắng, D[i] = 1 nếu cổng i trên đảo Gotland là Cổng Đi và = 0 nếu cổng i là Cổng Đến .

Example

Input:

4 5

3 5 2 5

4 4 4 1 3

Output:

0 1 1 0

1 0 1 1 0

* **Người up:** hard7771988
* **Điểm:** 0.58
* **Ngôn ngữ cho phép:**
* **Nguồn bài:** Dựa theo BOI 2001

Thống nhất đất nước ( ELECT )

**Giới hạn**

* **Thời gian:** 0.087s
* **Bộ nhớ:** 1536MB
* **Mã nguồn:** 20000 bytes

Sau nhiều năm chiến tranh liên miên giữa các Đảng phái , nước X rơi vào tình trạng đói nghèo , người dân khổ cực trăm bề . Nhận thức được tiếp tục kéo dài chiến tranh sẽ càng bất lợi cho đất nước , các Đảng trong nước X đã quyết định họp bàn nhau lại , bỏ qua hiềm khích chung để xây dựng lại đất nước. Việc làm đầu tiên sẽ là họp để chọn ra các vị đại biểu để lập nên Quốc Hội . Mỗi Đảng đã chọn ra 2 gương mặt tiêu biểu nhất cho Đảng của mình để ứng cử vào Quốc Hội . Tuy nhiên trong số các vị đại biểu của các Đảng này thì có một số vị vì lý do cá nhân trong chiến tranh nên rất căm thù nhau ( ví dụ như là ông A của Đảng P ghét ông B của Đảng Q … ) . Vì lý do chính trị mà trong Quốc Hội mỗi Đảng chỉ được phép có một người mà thôi . Ngoài ra để đảm bảo Quốc Hội làm việc 1 cách công minh thì các vị đại biểu Quốc Hội phải được chọn ra sao cho đảm bảo không có ai thù ghét ai cả nếu không rất có thể chiến tranh sẽ lại nổ ra . Bạn là một người yêu chuộng hoà bình đồng thời là 1 lập trình viên siêu hạng . Bạn hãy xem xét xem liệu có 1 cách tổ chức Quốc Hội sao cho thoả mãn được các yêu cầu đề ra hay không ?

Input

Dòng 1 : 2 số nguyên N và M ( 1 ≤ N≤ 8000 , 1 ≤ M ≤ 20000 ) tương ứng là số Đảng và só mối quan hệ thù ghét nhau giữa các thành viên của các Đảng . ( Các thành viên của Đảng 1 có số hiệu là 1 , 2 ; các thành viên của Đảng 2 có số hiệu là 3 , 4 … Thành viên của Đảng i sẽ có số hiệu là i\*2-1 và i\*2 ) .   
M dòng tiếp theo mỗi dòng gồm 2 số nguyên u , v cho biết người u và người v ghét nhau . ( 1 ≤ u < v ≤ N\*2 ) .

Output

Dòng 1 : Ghi 0 nếu không có phương án thoả mãn và 1 nếu có phương án thoả mãn.   
Nếu dòng 1 là 1 thì dòng thứ 2 ghi ra N số nguyên là số hiệu của các thành viên được chọn vào Quốc Hội .

Example

Input:

3 2

1 3

2 4

Output:

1

1 4 5

* **Người up:** hard7771988
* **Điểm:** 0.88
* **Ngôn ngữ cho phép:**
* **Nguồn bài:** Base on problem of Wojciech Rytter

Trải thảm đỏ ( CARPET )

**Giới hạn**

* **Thời gian:** 0.1s
* **Bộ nhớ:** 1536MB
* **Mã nguồn:** 20000 bytes

Sảnh đường lớn của khối chuyên Sư Phạm đã được xây dựng để dành cho lễ kỷ niệm 40 năm thành lập trường . Nền nhà của sảnh đường có kích thước là 2000\*2000 mét vuông . Và để lát nền cho sảnh đường người ta cần lát các tấm gạch có kích thước là 1 mét vuông . Thật không may là gần đến ngày kỷ niệm trường rồi thì do sơ ý của 1 số công nhân đã làm đổ sơn ra nền , làm hỏng mất vẻ đẹp của sảnh đường . Tuy nhiên thật may mắn là vẫn có thể mua thảm cùng màu về để che đi những chỗ bị dính sơn .   
Có một điều rất kỳ thú đó là vết sơn đó lại có hình là một cung của hình tròn ( có tâm nằm trong phạm vi nền nhà ) . Bên thi công muốn sửa chữa lỗi lầm bằng cách mua một tấm thảm có dạng là một hình chữ nhật và sau đó trải lên sàn nhà che đi vểt sơn sao cho cạnh của tấm thảm sau khi trải sẽ trùng với các cạnh của các viên gạch ( tức là các góc của tấm thảm sẽ phải có toạ độ nguyên ) . Điều đau đầu của bên thi công giờ là không biết mua tấm thảm với kích thước bao nhiêu thì là có thể lấp được hết vết sơn mà diện tích miếng thảm đó lại là nhỏ nhất . Bạn hãy lập trình giúp bên thi công giải bài toán khó này . Vì bạn là lập trình viên rất giỏi nên bên thi công chỉ cho bạn biết toạ độ của 2 điểm đầu mút của cung và của 1 điểm nằm trên cung tròn đó mà thôi .   
  
Chú ý bài này nên làm cẩn thận , suy xét hết các trường hợp có thể xảy ra .

Input

Dòng 1 : Số nguyên dương T là số bộ test ( 1 ≤ T ≤ 45 ) . T nhóm dòng tiếp theo , mỗi nhóm dòng mô tả 1 bộ test , có format như sau :   
2 dòng đầu , mỗi dòng gồm 2 số nguyên là toạ độ của 2 điểm đầu mút của vết sơn .   
Dòng thứ 3 gồm 2 số nguyên là toạ độ của 1 điểm nằm trên vết sơn đó . Biết rằng các toạ độ đều là số nguyên có trị tuyệt đối ≤ 1000 và cung tròn nằm trong hình vuông [-1000,1000] \* [-1000,1000] .

Output

Với mỗi test ghi ra 1 số nguyên X là diện tích nhỏ nhất của miếng thảm .

Example

Input:

1

476 612

487 615

478 616

Output:

66

* **Người up:** hard7771988
* **Điểm:** 1.27
* **Ngôn ngữ cho phép:**
* **Nguồn bài:** Base on a problem of Alexander Mironenko

Số học 1 ( JACOBI )

**Giới hạn**

* **Thời gian:** 0.603s
* **Bộ nhớ:** 1536MB
* **Mã nguồn:** 20000 bytes

Tìm tất cả số các nguyên x thoả mãn (x\*x) mod n = a mod n . Trong đó n là số nguyên tố và ước chung lớn nhất của a và n = 1 , 0 ≤ x ≤ n – 1 .

Input

Dòng 1 : số nguyên K là số bộ test ( 1 ≤ K ≤ 100000 ) . K dòng tiếp theo mỗi dòng gồm 2 số nguyên a , n ( 1 ≤ a , n ≤ 32767 ) .

Output

Với mỗi test ghi ra tất cả các số nguyên x thoả mãn theo thứ tự tăng dần trên 1 dòng . Nếu không có số nguyên x nào thoả mãn thì ghi ra “Khong co” .

Example

**Input:**

5

4 17

3 7

2 7

14 31

10007 20011

**Output:**

2 15

Khong co

3 4

13 18

5382 14629

* **Người up:** hard7771988
* **Điểm:** 0.78
* **Ngôn ngữ cho phép:**
* **Nguồn bài:** Base on a problem of Michael Medvedev

Chu vi các hình chữ nhật ( RECTP )

**Giới hạn**

* **Thời gian:** 0.079s
* **Bộ nhớ:** 1536MB
* **Mã nguồn:** 20000 bytes

Cho N hình chữ nhật có các cạnh song song với các trục Ox và Oy .   
Hãy tính chu vi của hình tạo bởi N hình chữ nhật này . Với định nghĩa chu vi của N HCN là tổng độ dài các đường biên giúp phân biệt đâu là miền nằm trong các HCN và miền nằm ngoài các HCN .   
Chú ý : N hình chữ nhật này có thể tách rời nhau , không nhất thiết là đè lên nhau .

Input

Dòng 1 : số nguyên N ( 1 ≤ N ≤ 10000 ) .   
N dòng tiếp theo , mỗi dòng gồm 4 số nguyên x1 , y1 , x2 , y2 tương ứng là toạ độ góc trái dưới và góc phải trên của hình chữ nhật thứ i .( 0 ≤ x1 ≤ x2 ≤ 30000 , 0 ≤ y1≤ y2 ≤ 30000 ) .

Output

Gồm 1 dòng ghi ra chu vi của N hình chữ nhật .

Example

Input:

2

10 10 20 20

15 15 25 30

Output:

70

Giải thích test ví dụ : Đó là tổng độ dài các đoạn tô đậm trong hình vẽ .

* **Người up:** hard7771988
* **Điểm:** 0.90
* **Ngôn ngữ cho phép:**
* **Nguồn bài:** IOI

Diện tích các tam giác vuông cân ( TRIANGLE )

**Giới hạn**

* **Thời gian:** 0.103s
* **Bộ nhớ:** 1536MB
* **Mã nguồn:** 20000 bytes

Cho N tam giác vuông cân . Hãy tính diện tích miền bị phủ bởi N tam giác này .

Input

Dòng 1 : số nguyên N ( 1 ≤ N ≤ 2000 ) .   
N dòng tiếp theo , mỗi dòng gồm 3 số nguyên xi , yi , mi ( -10^7 ≤ xi , yi ≤ 10^7 , 1 ≤ mi ≤ 1000 ) mô tả toạ độ tam giác thứ i , 3 đỉnh tam giác i sẽ có toạ độ (xi,yi) , (xi+mi,yi) , (xi,yi+mi) .

Output

Gồm 1 dòng duy nhất ghi ra diện tích miền bị phủ .

Example

Input:

5

-5 -3 6

-1 -2 3

0 0 2

-2 2 1

-4 -1 2

Output:

24.5

* **Người up:** hard7771988
* **Điểm:** 0.72
* **Ngôn ngữ cho phép:**
* **Nguồn bài:** BOI 2002

© VNOI Team 2015

Khoảng cách mong manh ( POINT )

**Giới hạn**

* **Thời gian:** 0.43s
* **Bộ nhớ:** 1536MB
* **Mã nguồn:** 50000 bytes

Cho N điểm phân biệt trên mặt phẳng toạ độ . Toạ độ của điểm i là ( Xi , Yi ) trong đó Xi , Yi là các số nguyên ( -10000 ≤ Xi , Yi ≤ 10000 ) .   
Ta định nghĩa khoảng cách giữa 2 điểm (X1,Y1) , (X2,Y2) là khoảng cách Manhattan được tính = | X1 – X2 | + | Y1 – Y2 | .   
Hàm Q(X,Y) := | X – X1 | + | X – X2 | + … + | X – Xn | + | Y – Y1 | + … |Y – Yn | .   
( Trong đó X , Y là 2 số nguyên thoả mãn -10000 ≤ X , Y ≤ 10000 và Xi ≠ X hoặc Yi ≠ Y với mọi i = 1 .. n ) .   
Hãy tìm tập tất cả các điểm nguyên (X,Y) để hàm Q(X,Y) có giá trị nhỏ nhất .

Input

Dòng 1 : số nguyên dương T là số bộ test ( T ≤ 20 ) .   
Các nhóm dòng sau mô tả 1 bộ test . 1 bộ test sẽ có format như sau :   
Dòng 1 : số nguyên dương N ( N ≤ 10000 ) .   
N dòng tiếp theo , dòng thứ i gồm 2 số nguyên là toạ độ của điểm thứ i .

Output

Với mỗi bộ test ghi 1 dòng gồm 2 số nguyên dương S , K tương ứng là giá trị nhỏ nhất của hàm Q(X,Y) và số lượng điểm thoả mãn yêu cầu .

Example

Input:

1

2

0 1

1 0

Output:

2 2

* **Người up:** hard7771988
* **Điểm:** 1.05
* **Ngôn ngữ cho phép:**
* **Nguồn bài:** Dựa theo 1 bài dễ hơn của USACO

Báo động đỏ ( ALERT )

**Giới hạn**

* **Thời gian:** 0.1s
* **Bộ nhớ:** 1536MB
* **Mã nguồn:** 50000 bytes

Thế giới những năm 2077 hình thành nên 2 thái cực rõ ràng , các nước hoặc là đi theo con đường Chủ Nghĩa Xã Hội hoặc là theo Tư Bản Chủ Nghĩa. Khởi xướng nên các luồng tư tưởng này là 2 nước Lào và Campuchia . Lào và 1 số nước thân Lào theo đường lối Xã Hội Chủ Nghĩa còn Campuchia và 1 số nước thân Campuchia theo Tư Bản Chủ Nghĩa. Như ta đã biết nền kinh tế các năm trong tương lai là nền kinh tế tri thức và của các mối quan hệ. Nếu trước đây 2 nước X và Y có quan hệ kinh tế là Z tỉ đôla với nhau và giờ X theo CNXH còn Y theo TBCN thì 2 nước này sẽ cắt đứt mối quan hệ kinh tế với nhau , đối với nền kinh tế thế giới thì thực sự là 1 tổn thất lớn , còn nếu 2 nước cùng đi theo cùng 1 con đường chính trị thì mối quan hệ đó vẫn được duy trì . Tuy nhiên năm nay mới là năm 2007 vì thế mới chỉ có Lào , các nước thân Lào là theo CNXH và Campuchia và các nước thân Campuchia theo TBCN , còn lại các nước vẫn theo con đường trung lập và tới năm 2077 họ mới chọn TBCN hay là XHCN.   
Biết bạn rất giỏi lập trình , các chuyên gia thuộc Liên Hợp Quốc muốn nhờ bạn hãy lập trình tính xem tới năm 2077 thì trong tình huống tốt nhất thì Tổng Giá Trị Kinh Tế Toàn Cầu là bao nhiêu ? Biết rằng Tổng Giá Trị Kinh Tế Toàn Cầu được tính bằng tổng giá trị các mối quan hệ kinh tế giữa các nước trên thế giới .

Input

Dòng 1 : Số nguyên dương N ( 1 ≤ N ≤ 200 ) là số lượng các quốc gia trên thế giới , các quốc gia được đánh số thứ tự từ 1 -> N .   
Dòng 2 : Số nguyên dương L là các nước tính tới thời điểm hiện tại đang theo CNXH .   
Dòng 3 : Gồm L số nguyên dương là chỉ số của các nước đang theo CNXH .   
Dòng 4 : Số nguyên dương C là các nước tính tới thời điểm hiện tại đang theo TBCN.   
Dòng 5 : Gồm C số nguyên dương là chỉ số của các nước đang theo TBCN .   
Dòng 6 : Số nguyên dương M ( 1 ≤ M ≤ N\*(N-1)/2 ) là số quan hệ kinh tế giữa các nước trên thế giới .   
M dòng tiếp theo , dòng thứ i gồm 3 số nguyên dương Xi Yi Zi ( 1 ≤ Xi ≠ Yi ≤ N , 1 ≤ Zi ≤ 1000 ) mô tả 1 mối quan hệ kinh tế .

Output

Dòng 1 : Số nguyên dương K là Tổng Giá Trị Kinh Tế Toàn Cầu trong tình huống tốt nhất và số nguyên dương T là số nước theo XHCN trong tình huống đó .   
Dòng 2 : Ghi ra chỉ số của T nước theo CNXH trong tình huống tốt nhất đó. Nếu có nhiều phương án thì chỉ ra phương án mà có số lượng nước theo CNXH là nhiều nhất .

Example

Input:

3

1

1

1

3

1

1 2 10

Output:

10 2

1 2

* **Người up:** hard7771988
* **Điểm:** 1.13
* **Ngôn ngữ cho phép:**

Chuỗi hạt ( CHUOIHAT )

**Giới hạn**

* **Thời gian:** 0.173s
* **Bộ nhớ:** 1536MB
* **Mã nguồn:** 50000 bytes

Khi tiến hành khai quật khảo cổ ở một vương quốc xa xưa nọ, các nhà khoa học khai quật được rất nhiều chuỗi hạt lạ. Sau khi quan sát, các nhà khoa học thấy rằng các chuỗi hạt có một số đặc điểm chung.

Mỗi chuỗi hạt là một sợi dây được đính các hạt ngọc làm bằng một chất liệu cổ xưa. Các chuỗi hạt đều có số lượng hạt ngọc bằng nhau. Hơn nữa, mỗi hạt ngọc là một hình cầu có đường kính là một số nguyên dương, và nếu lần từ trái sang phải trên chuỗi hạt, người ta thấy các hạt ngọc có đường kính tăng dần. Nếu đánh số vị trí các hạt ngọc bắt đầu từ 1, theo thứ tự từ trái sang phải, người ta nhận thấy rằng hạt ngọc thứ i có đường kính không vượt qúa 2i. Các nhà khoa học cho rằng, dân tộc cổ xưa này hẳn đã làm ra tất cả các chuỗi hạt có cùng những đặc điểm này, dù chúng hiện còn đang rải rác ở đâu đó trên trái đất.

Sau đó không lâu, các nhà khoa học tìm ra một mảnh da, trên đó có ghi một con số theo loại chữ số cổ xưa. Họ cho rằng mảnh da này có liên quan đến các chuỗi hạt kỳ lạ nọ. Sau nhiều cố gắng, các nhà khoa học đã đưa được con số trên mảnh da về hệ chữ số thập phân, và ký hiệu là X.

Manh mối đến đây thì dừng lại, vì các nhà khoa học không tìm thấy được vết tích nào khác nữa, và cũng không tìm ra được mối quan hệ giữa X và các chuỗi hạt.

Đến đây, một nhà khoa học người Việt đề nghị, hãy thử xác định chuỗi hạt có thứ tự từ điển là X, biết đâu đây sẽ là manh mối?

Yêu cầu

Bạn hãy viết chương trình giúp nhà khoa học xác định chuỗi hạt có thứ tự từ điển là X.

Dữ liệu

* Dòng 1: chứa số nguyên dương N, là số hạt ngọc trong mỗi chuỗi hạt
* Dòng 2: chứa số nguyên dương X

Kết qủa

Gồm 1 dòng duy nhất, chứa N số nguyên, cách nhau một khoảng trắng, xác định chuỗi hạt có thứ tự từ điển là X.

(để biểu diễn một chuỗi hạt, cần in ra N số nguyên tương ứng là đường kính của các hạt ngọc trong chuỗi hạt, theo thứ tự từ trái sang phải)

Giới hạn

* N là số nguyên dương trong phạm vi [1, 250]
* X là số nguyên dương trong phạm vi từ 1 đến số lượng tối đa các chuỗi hạt.

Ví dụ

Dữ liệu

2

4

Kết qủa

2 3

Giải thích

Các chuỗi hạt sắp theo thứ tự từ điển:

1 2, 1 3, 1 4, 2 3, 2 4

Chuỗi hạt thứ 4 là 2 3

* **Người up:** paulmcvn
* **Điểm:** 0.37
* **Ngôn ngữ cho phép:**
* **Nguồn bài:** Thầy Nguyễn Thanh Hùng - PTNK

Lịch thi đấu bóng đá ( BONGDA )

**Giới hạn**

* **Thời gian:** 0.054s
* **Bộ nhớ:** 1536MB
* **Mã nguồn:** 50000 bytes

Một giải thi đấu bóng đá được tổ chức theo thể thức thi đấu vòng tròn, ở đó hai đội bóng bất kỳ sẽ được thi đấu với nhau đúng một trận.

Trong mỗi trận đấu, đội thắng được 3 điểm, còn đội thua được 0 điểm. Không có kết qủa hòa (nếu hòa trong hai hiệp chính thì hai đội sẽ phải thi đấu luân lưu để phân định thắng thua).

Kết thúc mùa giải, đội vô địch là đội có số điểm cao nhất. Trong trường hợp có nhiều hơn một đội được điểm cao nhất thì các đội này đồng đoạt chức vô địch.

Giải đã diễn ra được một thời gian. Lúc này một bài toán nảy sinh: người ta cần biết mỗi đội bóng còn có khả năng đoạt chức vô địch nữa hay không?

Yêu cầu

Biết kết qủa các trận đấu đã diễn ra, bạn hãy cho biết mỗi đội bóng còn có khả năng đoạt chức vô địch mùa giải nữa hay không?

Dữ liệu

* Dòng 1: chứa số nguyên dương N là số đội bóng tham gia giải đấu. Các đội bóng được đánh số thứ tự từ 1 đến N.
* N dòng tiếp theo, dòng i chứa N số nguyên dương a i1, a i2, ... , a iN, trong đó a ijcho biết trạng thái của trận đấu giữa đội i và đội j:
  + a ij= 0, nếu đội j thắng.
  + a ij= 1 nếu đội i thắng
  + a ij= 2 nếu trận đấu chưa diễn ra.

Nếu i khác j, dữ liệu vào đảm bảo a ij= a ji= 2 hoặc a ij+ a ji= 1. Dữ liệu vào đảm bảo a ii= 0 với mọi i

Kết qủa

Gồm 1 dòng duy nhất, là một dãy nhị phân độ dài N, ở đó bit thứ i bằng 1 nếu đội bóng i còn khả năng vô địch mùa giải, và bằng 0 nếu đội bóng i không còn khả năng vô địch mùa giải.

Giới hạn

N là số nguyên dương trong phạm vi [1, 30]

Ví dụ

Dữ liệu

5

0 2 0 2 1

2 0 0 1 0

1 1 0 2 1

2 0 2 0 2

0 1 0 2 0

Kết qủa

10110

* **Người up:** paulmcvn
* **Điểm:** 0.51
* **Ngôn ngữ cho phép:**
* **Nguồn bài:** Thầy Nguyễn Thanh Hùng - PTNK

Nhãn của cây ( ITREE )

**Giới hạn**

* **Thời gian:** 0.023s
* **Bộ nhớ:** 1536MB
* **Mã nguồn:** 50000 bytes

Cho đồ thị cây có trọng số gồm N đỉnh , các đỉnh được đánh số từ 1 -> N . Gốc của cây là đỉnh 1 . Cha của đỉnh u là 1 đỉnh có số hiệu nhỏ hơn u . Mỗi đỉnh có một nhãn là 1 số thực A[i] . Trong đó nhãn của đỉnh 1 bằng 1 và nhãn của đỉnh lá bằng 0 . Biết rằng A[v] ≤ A[u] nếu v là con của u .   
Giá trị của 1 cây = Tổng ( ( A[u] – A[v] ) \* Trọng số cạnh (u,v) , với u là cha của v )   
Bây giờ người ta cho biết các cạnh của đồ thị và trọng số của các cạnh này nhưng không cho biết các A[i]. Hãy tính xem giá trị của cây thấp nhất là bao nhiêu.

Input

Dòng 1 là số nguyên T là số bộ test . ( 1 ≤ T ≤ 50 ) . T nhóm dòng tiếp theo mô tả từng bộ test . Mỗi bộ test sẽ có cấu trúc như sau :   
Dòng 1 : số nguyên dương N ( 1 ≤ N ≤ 1000 ) .   
Từ dòng 2 -> dòng N : dòng thứ i gồm 2 số nguyên dương u và c ( 1 ≤ u < i , 0 ≤ c ≤ 1000 ) cho biết cha của nút i là nút u và cạnh nối (u,i) có trọng số là c .

Output

Với mỗi test ghi ra giá trị thấp nhất có thể đạt được của cây trên 1 dòng với độ chính xác là 2 chữ số sau dấu chấm.

Example

Input:

1

4

1 1

1 2

2 1

Output:

3.00

Giải thích : Phương án tối ưu là A[1] = 1 , A[2] = 0.5 , A[3] = 0 , A[4] = 0 .

* **Người up:** hard7771988
* **Điểm:** 0.27
* **Ngôn ngữ cho phép:**

Đoạn cao trào của bản nhạc ( THEME )

**Giới hạn**

* **Thời gian:** 0.23s
* **Bộ nhớ:** 1536MB
* **Mã nguồn:** 50000 bytes

Trong một bản nhạc thường có những đoạn nhạc mà tác giả sử dụng nó nhiều lần ( ít nhất 2 lần ). Những đoạn đó gọi là "đoạn cao trào". Do có thể sử dụng nhiều giọng khác nhau ( son, la, si...) nên nốt đầu tiên của các lần xuất hiện có thể khác nhau, nhưng chệnh lệnh độ cao giữa hai nốt liên tiếp thì chắc chắn giống.   
VD: hai đoạn sau   
1 2 5 4 10   
và   
4 5 8 7 13   
được coi là một đoạn cao trào, vì chúng cùng sự chênh lệch độ cao : +1,+3,-1,+6   
Cho một bản nhạc, yêu cầu tìm độ dài đoạn cao trào dài nhất.   
+ Đoạn cao trào phải có từ 5 nốt nhạc trở lên.   
+ Những lần xuất hiện của đoạn không được chồng lên nhau ( không có nốt nhạc chung ).

Input

Dòng 1 : n = số nốt nhạc <= 5000   
Một số dòng sau là n nốt nhạc, mỗi nốt được quy ra số tự nhiên trong phạm vi 1..88.

Output

1 dòng chứa 1 số duy nhất là độ dài đoạn cao trào dài nhất. Nếu không tìm được đoạn nhạc nào, in ra 0.

Example

Input:

30

25 27 30 34 39 45 52 60 69 79 69 60 52 45 39 34 30 26 22 18

82 78 74 70 66 67 64 60 65 80

Output:

5

(5 nốt cuối dòng 1 và 5 nốt đầu dòng hai cùng là một đoạn)

* **Người up:** hard7771988
* **Điểm:** 0.14
* **Ngôn ngữ cho phép:**
* **Nguồn bài:** USACO , thanks Hoanh Tien

Chơi bi-a 1 lỗ ( CHEAT )

**Giới hạn**

* **Thời gian:** 0.579s
* **Bộ nhớ:** 1536MB
* **Mã nguồn:** 50000 bytes

Mike chơi bi-a 1 lỗ rất giỏi nên kiếm được rất nhiều tiền độ banh . Nhà Cái mất nhiều tiền vì Mike lắm nên cú lắm nên họ quyết phải kiểm tra xem liệu Mike có chơi gian hay không ?   
Thể thức chơi bi-a 1 lỗ là như này : Có N viên bi được đánh số từ 1 -> N , đặt trên bàn , người chơi phải đánh sao cho các viên bi này lọt lỗ theo đúng thứ tự từ 1 -> N . Viên i sẽ phải vào lỗ trước viên i+1 . Để kiểm tra Mike , nhà Cái thuê 1 tay thám tử . Tay thám tử này sẽ kiểm tra bằng cách là thỉnh thoảng lại tiến lại cái lỗ và bốc lên viên ở trên cùng trong lỗ . Sau khi Mike đã đánh hết các bi vào lỗ rồi thì thám tử sẽ bốc hết các viên ở trong lỗ ra từ viên trên cùng tới viên dưới cùng . Hãy giúp thám tử xác định xem liệu Mike có chơi gian không ? ( Xem test ví dụ để hiểu rõ hơn ) .   
  
Download test và solution tại [đây](https://vn.spoj.pl/SPOJVN/content/cheat.zip)

Input

Dòng 1 : số nguyên dương N ( N ≤ 100000 ) . N dòng tiếp theo mỗi dòng gồm 1 số nguyên ghi ra số chỉ trên trái bi mà thám tử lần lượt bốc lên được .

Output

Nếu xác định được Mike chơi gian thì ghi ra “YES” , ngược lại ghi “NO” .

Example

Input:

3

3

1

2

Output:

YES

Giải thích : Khi thám tử bốc được bi số 3 lên thì có nghĩa là bi số 1 , 2 đã vào lỗ rồi . Và như vậy bi trên cùng sau khi bốc bi số 3 ra phải là bi số 2 nhưng thám tử lại bốc ra được bi số 1 -> vô lý -> Mike đã ăn gian .

Input:

6

1

3

5

6

4

2

Output:

NO

Có thể xảy ra trường hợp thám tử bốc viên bi 1, 3, 5 ngay khi Mike vừa đánh chúng vào lỗ. Sau đó thám tử bốc những viên bi còn lại. Do đó không khẳng định được Mike đã ăn gian!

* **Người up:** hard7771988
* **Điểm:** 0.09
* **Ngôn ngữ cho phép:**
* **Nguồn bài:** Alex Samsonov

Cô gái chăn bò ( COWGIRL )

**Giới hạn**

* **Thời gian:** 0.284s
* **Bộ nhớ:** 1536MB
* **Mã nguồn:** 10000 bytes

Trên một thảo nguyên nhỏ bé có 1 gia đình gồm 3 anh em: 2 người anh trai là Nvutri và Andorea còn người em gái là Lola. Cuộc sống gia đình khá giả nhưng gia đình có truyền thống chăn nuôi và muốn để các con tự lập nên cha mẹ 3 người quyết định để các con hằng ngày sẽ đi chăn 1 số bò nào đó (tùy ý 3 người con).

Thảo nguyên là 1 cánh đồng chia làm M\*N ô vuông, mỗi con bò chỉ đứng trong 1 ô và mỗi ô chỉ chứa 1 con bò.Chỉ có 1 quy tắc duy nhất là không bao giờ được để 4 con bò tạo thành 1 hình vuông 2\*2 hoặc để trống 1 khu đất 2\*2.

Hai người anh mải chơi nên đã hối lộ kem để Lola chăn bò 1 mình. Lola muốn biết tất cả có bao nhiêu cách xếp bò thỏa mãn quy tắc trên để đề phòng mọi trường hợp. Vì con số này rất lớn nên hãy giúp Lola tính toán con số này.

Input

Dòng đầu gồm 1 số T duy nhất là số test (T ≤ 111)

T dòng tiếp theo gồm 2 số M, N cho biết kích thước của thảo nguyên (M\*N ≤ 30)

Output

Gồm T dòng, mỗi dòng ứng với 1 test là số cách xếp bò của test đó.

Example

Input:

1

1 1

Output:

2

* **Người up:** hard7771988
* **Điểm:** 0.12
* **Ngôn ngữ cho phép:**
* **Nguồn bài:** Thank to Nguyễn Trần Nam Khánh and Nguyễn Hoàng Nghĩa

Cân thăng bằng ( RTF )

**Giới hạn**

* **Thời gian:** 0.173s
* **Bộ nhớ:** 1536MB
* **Mã nguồn:** 5000 bytes

Có 1 bộ N loại quả cân , loại quả i nặng A[i] kg . Một bộ con của bộ N quả cân gồm k loại quả cân là A[i1] , A[i2] , … A[ik] gọi là thay thế được bộ N loại quả cân trên nếu như tồn tại hệ đẳng thức sau :   
A[1] = A[i1] \* H[1][1] + … + A[ik] \* H[1][k]   
A[2] = A[i1] \* H[2][1] + … + A[ik] \* H[2][k]   
…   
A[N] = A[i1] \* H[N][1] + … + A[ik]\* H[N][k]   
Trong đó H[i][j] là các số nguyên .   
Hãy tìm bộ quả cân thay thế được cho bộ N quả cân với số phần tử là nhỏ nhất .

Input

Dòng 1 là số nguyên T là số bộ test ( 1 ≤ T ≤ 100 ) . Các dòng tiếp theo mô tả T bộ test , mỗi bộ test gồm 2 dòng :   
Dòng 1 : Số nguyên dương N ( N ≤ 100 ) .   
Dòng 2 : N số nguyên dương A[1] , … , A[N] ( A[i] ≤ 60000 )

Output

Với mỗi bộ test ghi ra 1 số nguyên dương là số lượng quả cân của nhóm tìm được .

Example

Input:

1

3

2 3 4

Output:

2

* **Người up:** hard7771988
* **Điểm:** 0.83
* **Ngôn ngữ cho phép:**

Số không (I) ( VN\_ZR\_I )

**Giới hạn**

* **Thời gian:** 0.063s
* **Bộ nhớ:** 1536MB
* **Mã nguồn:** 50000 bytes

Lần đầu tiên được tiếp xúc với các vấn đề về cơ sở tin học, các học sinh đều ngỡ ngàng và thú vị khi được làm quan với hệ đếm cơ số 2.

Bài tập về nhà là mỗi người tự chọn cho mình một số nguyên N và viết các số 1, 2, 3, …, N dưới dạng nhị phân. Qua bài tập này, thầy giáo muốn biết:

* Học sinh đã nắm được cách biểu diễn nhị phân hay chưa.
* Đánh giá được mức độ ham mê tin học sinh trong lớp qua số N được chọn và cách trình bày bài làm.

1

1 0

11

1 0 0

1 0 1

11 0

111

1 0 0 0

1 0 01

1 0 10

1 0 11

11 0 0

11 0 1

111 0

1111

1 0 0 0 0

1 0 0 0 1

1 0 01 0

1 0 011

1 0 10 0

1 0 101

1 0 110

1 0 111

11 0 0 0

11 0 01

11 0 10

11 0 11

111 0 0

111 0 1

1111 0

11111

1 0 0 0 0 0

1 0 0 0 01

1 0 0 0 10

1 0 0 0 11

1 0 01 0 0

1 0 01 0 1

1 0 011 0

1 0 0111

1 0 10 0 0

1 0 10 0 1

1 0 101 0

1 0 1011

1 0 110 0

1 0 1101

1 0 1110

1 0 1111

11 0 0 0 0

11 0 0 0 1

11 0 01 0

11 0 011

11 0 10 0

11 0 101

11 0 110

11 0 111

111 0 0 0

Một bạn đã rất cố gắng thực hiện bài tập, chọn số N khá lớn, ghi các số từ 1 tới N dưới dạng nhị phân, mỗi số trên một dòng. Sau đó để cho bài làm có dạng hấp dẫn hơn, bạn học sinh đó chọn một số nguyên K lớn hớn 0 và ở mỗi dòng – tô đỏ các 0 thứ nhất, thứ K + 1, 2K + 1, … Ở hình trên, N = 56 và K = 2. Các số 0 màu đỏ được gạch dưới.

Các bạn trong lớp rất thích thú khi thấy bài làm này và định in để nộp. Nhưng có một bạn lo lắng: “Máy in màu của mình sắp hết mặc đỏ. Với N và K đã chọn, sẽ có bao nhiêu số 0 được viết bằng màu đỏ?”. Hãy giúp các bạn đang làm bài tập trả lời câu hỏi trên.

Input

Gồm nhiều dòng, mỗi dòng chứa 2 số nguyên N và K cách nhau ít nhất một dấu cách. (1 < N ≤ 2147483647; K > 0)

Output

Gồm nhiều dòng, mỗi dòng gồm 1 số là kết quả tìm được của từng test.

Example

Input:

4 1

56 2

Output:

3

74

* **Người up:** duonghanguyen
* **Điểm:** 0.41
* **Ngôn ngữ cho phép:**
* **Nguồn bài:** Thầy Nguyễn Thanh Tùng

HAOI 5000 ( HAOI5000 )

**Giới hạn**

* **Thời gian:** 0.079s
* **Bộ nhớ:** 1536MB
* **Mã nguồn:** 50000 bytes

Thể thức của cuộc thi tin học HAOI mỗi năm lại được thay đổi để không làm thí sinh cảm thấy nhàm chán. Chắc hẳn các bạn vẫn nhớ những cuộc thi HAOI 3000, 4000 đã được mô tả những lần trước. Đến lần thứ 5000, cuộc thi lại được xây dựng theo một phong cách khác hẳn. Phòng thi là một khu có dạng hình tròn với N máy tính đặt cách đều nhau. Máy thứ 1 được đặt giữa máy thứ N và máy thứ 2, máy thứ 2 được đặt giữa máy thứ 1 và máy thứ 3, cứ như vậy đến máy thứ N thì được đặt giữa máy N – 1 và máy thứ 1. Phòng thi có kích thước rất lớn những lại chỉ có một số thí sinh tham gia cuộc thi này thậm chí vấn đề còn trở nên phức tạp hơn khi có thể có nhiều thí sinh cùng thi trên một máy (kiểu như thi đồng đội). Khoảng cách giữa 2 máy tính được định nghĩa là khoảng cách để đi từ máy này đến máy kia (tính theo đường ngắn hơn). Nói cách khác, khoảng cách giữa hai máy u và v là min(|u – v|, N – |u – v|).

Giám thị MSN đang cần tìm một vị trí để có thể bao quát phòng thi tốt nhất. Vị trí quan sát tốt nhất là tại máy tính mà tổng khoảng cách của nó đến các máy đang thi là nhỏ nhất (nếu có một máy tính có nhiều thí sinh cùng dự thi thì khoảng cách đến máy đó phải được tính nhiều lần).

Input

* Dòng đầu tiên là hai số nguyên dương N và K lần lượt là tổng số máy tính có trong phòng thi và số thí sinh đang dự thi.
* Dòng tiếp theo ghi K số nguyên trong khoảng [1, n]. Số thứ u thể hiện số hiệu máy tính của thí sinh thứ u.

Output

* Dòng đầu tiên gồm một số là khoảng cách nhỏ nhất tìm được.
* Dòng thứ hai gồm một số là số lượng các máy tính có thể chọn
* Dòng thứ ba gồm một số số là danh sách số hiệu các máy tính có tổng khoảng cách tới các máy đang thi là nhỏ nhất

Giới hạn

* 1 ≤ N ≤ 10 6
* 1 < K ≤ 10 5

Ví dụ

Input:

20 10

1 1 18 5 6 14 7 4 8 9

Output:

35

2

5 6

* **Người up:** duonghanguyen
* **Điểm:** 0.34
* **Ngôn ngữ cho phép:**
* **Nguồn bài:** Based on problem HAOI5000 (author Pham Hai Minh - used in HAOI 2007). Thank to Ly Xuan Sang and Nguyen Thanh An.

Số học 2 ( NEWJ )

**Giới hạn**

* **Thời gian:** 0.35s
* **Bộ nhớ:** 1536MB
* **Mã nguồn:** 20000 bytes

Tìm tất cả số các nguyên x thoả mãn (x\*x) mod n = a mod n. Trong đó n là số nguyên tố và ước chung lớn nhất của a và n = 1, 0 ≤ x ≤ n – 1 .

Input

Dòng 1: số nguyên K là số bộ test ( 1 ≤ K ≤ 120000 ). K dòng tiếp theo mỗi dòng gồm 2 số nguyên a , n ( 1 ≤ a, n ≤ 1000000 ).

Output

Với mỗi test ghi ra tất cả các số nguyên x thoả mãn theo thứ tự tăng dần trên 1 dòng . Nếu không có số nguyên x nào thoả mãn thì ghi ra “Khong co”.

Example

**Input:**

5

4 17

3 7

2 7

14 31

10007 20011

**Output:**

2 15

Khong co

3 4

13 18

5382 14629

* **Người up:** hard7771988
* **Điểm:** 0.98
* **Ngôn ngữ cho phép:**
* **Nguồn bài:** Base on a problem of Michael Medvedev

Yugi-Oh ( YUGI )

**Giới hạn**

* **Thời gian:** 0.1s
* **Bộ nhớ:** 1536MB
* **Mã nguồn:** 50000 bytes

Các bạn đã đọc bộ truyện tranh Nhật Bản Yugi-oh chắc hẳn ai cũng cực kì yêu thích trò chơi bài Magic. Bộ bài và chiến thuật chơi quyết định đến sự thắng thua của đối thủ(mà sự thắng thua thì còn liên quan đến cả tính mạng >\_<). Vì thế tầm quan trọng của bộ bài là rất lớn. Một bộ bài tốt không chỉ bao gồm các quân bài mạnh mà còn phụ thuộc vào sự hỗ trợ tương tác giữa các quân bài. Bộ bài của Yugi là một bộ bài có sự bổ sung, hỗ trợ cho nhau rất tốt, điều này là 1 trong các nguyên nhân khiến Kaiba luôn là kẻ chiến bại.

Tình cờ Kaiba đã tìm được 1 quân bài ma thuật mà chức năng của nó là chia bộ bài hiện có của đối thủ ra làm K phần, mỗi phần có ít nhất 1 quân bài (điều này làm giảm sức mạnh của đối thủ). Kaiba quyết định áp dụng chiến thuật này với Yugi. Hiện tại Yugi có trong tay N quân bài, 2 quân bài i, j có sức mạnh tương tác a(i,j) (a(i,j) = a(j,i)). Kaiba muốn chia các quân bài thành K phần theo quy tắc sau:

* Giả sử K phần là P1, P2, ..., Pk thì độ giảm sức mạnh giữa 2 phần u,v là b(u,v) = min(a(i,j) với i thuộc Pu, j thuộc Pv).
* Độ giảm sức mạnh của bộ bài là S = min(b(u,v) với 1 ≤ u, v ≤ K).

Kaiba muốn chia K phần sao cho S lớn nhất

Input

* Dòng đầu là 2 số N,K(2 ≤ K ≤ N ≤ 200)
* N dòng tiếp theo mỗi dòng là N số a(i,j) (a(i,j) ≤ 32767; nếu i = j thì a(i,j) = 0)

Output

Gồm 1 dòng duy nhất là S lớn nhất

Example

Input:

4 3

0 1 2 3

1 0 2 3

2 2 0 3

3 3 3 0

Output:

2

* **Người up:** nghia
* **Điểm:** 0.22
* **Ngôn ngữ cho phép:**
* **Nguồn bài:** Không rõ nguồn gốc

Dãy con tăng dài nhất ( bản dễ ) ( LIQ )

**Giới hạn**

* **Thời gian:** 0.73s
* **Bộ nhớ:** 1536MB
* **Mã nguồn:** 20000 bytes

Cho một dãy số nguyên gồm N phần tử A[1], A[2], ... A[N].   
Biết rằng dãy con tăng đơn điệu là 1 dãy A[i 1],... A[i k] thỏa mãn   
i 1< i 2< ... < i kvà A[i 1] < A[i 2] < .. < A[i k]. Hãy cho biết dãy con tăng đơn điệu dài nhất của dãy này có bao nhiêu phần tử?   
  
Download test và solution (C/C++, Pascal) tại [đây](https://vn.spoj.pl/SPOJVN/content/liq.zip).

Input

* Dòng 1 gồm 1 số nguyên là số N (1 ≤ N ≤ 1000).
* Dòng thứ 2 ghi N số nguyên A[1], A[2], .. A[N] (1 ≤ A[i] ≤ 10000).

Output

Ghi ra độ dài của dãy con tăng đơn điệu dài nhất.

Ví dụ

Input:

6

1 2 5 4 6 2

Output:

4

Giải thích test ví dụ: Dãy con dài nhất là dãy A[1] = 1 < A[2] = 2 < A[4] = 4 < A[5] = 6, độ dài dãy này là 4.

Gợi ý: Sử dụng phương pháp Quy Hoạch Động. F[i]: Độ dài dãy con đơn điệu tăng dài nhất mà phần tử cuối cùng là số A[i] này.

* **Người up:** hard7771988
* **Điểm:** 0.03
* **Ngôn ngữ cho phép:**
* **Nguồn bài:** Dân gian

Phương trình Pythagore nghiệm nguyên ( PYTHAEQ )

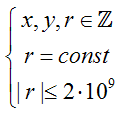
**Giới hạn**

* **Thời gian:** 0.337s
* **Bộ nhớ:** 1536MB
* **Mã nguồn:** 50000 bytes

Cho phương trình có dạng:

http://www.spoj.pl/content/duonghanguyen:PYTHAEQ_1.PNG

với



Hãy tìm số lượng nghiệm của phương trình đó

Input

Gồm một số r duy nhất

Output

Gồm một số duy nhất là đáp số của bài toán.

Example

Input 1:

1

Output 1:

4

Input 2:

3

Output 2:

4

Input 3:

10

Output 3:

12

* **Người up:** duonghanguyen
* **Điểm:** 0.44
* **Ngôn ngữ cho phép:**
* **Nguồn bài:** Một bài cố điển

Dãy con tăng dài nhất (bản khó) ( LIS )

**Giới hạn**

* **Thời gian:** 0.12s
* **Bộ nhớ:** 1536MB
* **Mã nguồn:** 50000 bytes

(Giống bài LIQ) Cho một dãy gồm N số nguyên (1 ≤ N ≤ 30000). Hãy tìm dãy con tăng dài nhất trong dãy đó. In ra số lượng phần tử của dãy con. Các số trong phạm vi longint.

Input

* Dòng đầu tiên gồm số nguyên N.
* Dòng thứ hai gồm N số mô tả dãy.

Output

Gồm một số nguyên duy nhất là đáp số của bài toán

Example

**Input:**

5

2 1 4 3 5

**Output:**

3

* **Người up:** duonghanguyen
* **Điểm:** 0.04
* **Ngôn ngữ cho phép:**
* **Nguồn bài:** Bài cổ điển

Công viên Disneyland (version 1) ( DISNEY1 )

**Giới hạn**

* **Thời gian:** 0.1s
* **Bộ nhớ:** 1536MB
* **Mã nguồn:** 50000 bytes

Công viên Disneyland khai trương N điểm vui chơi mới. Một hôm Bờm và Cuội đến chơi Công viên. Qua việc hỏi thăm người hướng dẫn Bờm và Cuội đã biết thời gian cụ thể để đi từ điểm vui chơi i đến địa điểm vui chơi j. Bờm và Cuội muốn đi thăm hết tất cả các địa điểm theo nguyên tắc sau:   
-Mỗi người đi từ điểm số 1, qua 1 số điểm và quay trở về điểm số 1.   
-Các điểm đến trên đường đi của Bờm và Cuội là 1 dãy số tăng dần( trừ điểm số 1 khi quay về).   
-Mỗi điểm vui chơi phải thuộc ít nhất 1 trong 2 đường đi của Bờm và Cuội.   
-Thời gian đi là nhỏ nhất (tính cả thời gian quay về 1).

Input

-Dòng đầu là số N (N<201)   
-Tiếp theo là ma trận N dòng N cột: 1 số nguyên không âm(<1001) trên hàng i cột j cho ta biết thời gian đi từ địa điểm i đến địa điểm j.   
(A[i][j] = A[j][i] , A[i][i] = 0 với mọi i, j).

Output

Gồm 1 số duy nhất là thời gian nhỏ nhất để Bờm và Cuội thăm quan Công Viên theo nguyên tắc trên.

Example

Input:

4

0 1 4 3

1 0 2 4

4 2 0 4

3 4 4 0

Output:

10

(Giải thích : cách đi tốt nhất là Bờm đi 1->2->3->4->1 và Cuội đứng yên)

* **Người up:** nghia
* **Điểm:** 0.28
* **Ngôn ngữ cho phép:**
* **Nguồn bài:** Thi vòng 2 nãm 2006 và cám ơn Mr.Nguyễn Trần Nam Khánh

Công viên Disneyland (version 2) ( DISNEY2 )

**Giới hạn**

* **Thời gian:** 0.1s
* **Bộ nhớ:** 1536MB
* **Mã nguồn:** 50000 bytes

Giống như bài DISNEY1, chỉ khác là mỗi điễm trong N điểm vui chơi chỉ thuộc 1 trong 2 đường đi của Bờm hoặc Cuội(tất nhiên là trừ điểm số 1).

Input

Giống như bài DISNEY1

Output

Giống như bài DISNEY1

Example

Input:

4

0 1 4 3

1 0 2 4

4 2 0 4

3 4 4 0

Output:

10

* **Người up:** nghia
* **Điểm:** 0.32
* **Ngôn ngữ cho phép:**
* **Nguồn bài:** Từ việc hiểu nhầm đề bài DISNEY1

Lại 1 bài phân việc ( COST )

**Giới hạn**

* **Thời gian:** 0.029s
* **Bộ nhớ:** 1536MB
* **Mã nguồn:** 50000 bytes

Có N chuyên viên lập trình và M công việc. Nếu chuyên viên thứ i mà làm j công việc thì sẽ tốn chi phí là C[i] \* j \* j .   
Người ta cho bạn N xâu ký tự , ký tự thứ j của xâu i là 'Y' tức là chuyên viên thứ i có thể làm được công việc thứ j và 'N' trong trường hợp ngược lại. Bạn hãy lập trình tính xem tổng chi phí phải trả nhỏ nhất là bao nhiêu ?

Input

Dòng 1 : số nguyên dương N ( 0 < N < 51 ) .   
Dòng 2 : N số nguyên dương là C[1] , ... , C[N] ( 0 < C[i] < 50000 ) . N dòng tiếp theo , mỗi dòng gồm 1 xâu M ký tự mô tả như ở trên. ( 0 < M < 41 ) .

Output

Gồm 1 số nguyên duy nhất là chi phí nhỏ nhất. Trong trường hợp không thể hoàn thành M công việc này được thì ghi ra -1 .

Ví dụ

Input:

3

2 3 4

YYN

YNY

NNN

Output:

11

Giải thích test ví dụ : Người thứ 1 làm 2 công việc là 1 và 2 mất chi phí là 2 \* 2 \* 2 = 8 , người thứ 2 làm công việc 3 mất chi phí là 3 . Tổng chi phí sẽ là 11 .

* **Người up:** hard7771988
* **Điểm:** 0.96
* **Ngôn ngữ cho phép:**

Số không (II) ( VN\_ZR\_II )

**Giới hạn**

* **Thời gian:** 0.083s
* **Bộ nhớ:** 1536MB
* **Mã nguồn:** 50000 bytes

Đề bài giống với bài Số không (I) nhưng có một điểm hơi khác:

Lần đầu tiên được tiếp xúc với các vấn đề về cơ sở tin học, các học sinh đều ngỡ ngàng và thú vị khi được làm quan với hệ đếm cơ số 2.

Bài tập về nhà là mỗi người tự chọn cho mình một số nguyên N và viết các số 1, 2, 3 , …, N dưới dạng nhị phân. Qua bài tập này, thầy giáo muốn biết:

* Học sinh đã nắm được cách biểu diễn nhị phân hay chưa.
* Đánh giá được mức độ ham mê tin học sinh trong lớp qua số N được chọn và cách trình bày bài làm.

Một bạn đã rất cố gắng thực hiện bài tập, chọn số N khá lớn, ghi các số từ 1 tới N dưới dạng nhị phân, mỗi số trên một dòng. Sau đó để cho bài làm có dạng hấp dẫn hơn, bạn học sinh đó chọn một số nguyên K lớn hớn 0 và ở mỗi dòng, xét từng nhóm được tạo bởi dãy dài nhất các chữ số 0 liên tiếp – mỗi nhóm tô đỏ các 0 thứ nhất, thứ K + 1, 2K + 1, …

Các bạn trong lớp rất thích thú khi thấy bài làm này và định in để nộp. Nhưng có một bạn lo lắng: “Máy in màu của mình sắp hết mặc đỏ. Với N và K đã chọn, sẽ có bao nhiêu số 0 được viết bằng màu đỏ? ”.Hãy giúp các bạn đang làm bài tập trả lời câu hỏi trên.

Input

Gồm nhiều dòng, mỗi dòng chứa 2 số nguyên N và K cách nhau ít nhất một dấu cách. (1 < N ≤ 2147483647; K > 0)

Output

Gồm nhiều dòng, mỗi dòng gồm 1 số là kết quả tìm được của từng test.

Example

Input:

4 1

56 2

Output:

3

86

* **Người up:** duonghanguyen
* **Điểm:** 1.13
* **Ngôn ngữ cho phép:**
* **Nguồn bài:** Từ việc hiểu nhầm đề bài Số không (I)

Truy vấn trên cây ( QTREEX )

**Giới hạn**

* **Thời gian:** 0.854s
* **Bộ nhớ:** 1536MB
* **Mã nguồn:** 50000 bytes

Cho một cây gồm N nút đánh số từ 1->N. Các cạnh của cây đánh số từ 1->N-1, mỗi cạnh có trọng số là một số nguyên. Bạn cần viết chương trình thực hiện dãy các lệnh sau:   
CHANGE i v => Thay đổi trọng số của cạnh thứ i thành v   
NEGATE a b => Đảo dấu trọng số của tất cả các cạnh nằm trên đường đi từ a đến b   
QUERY a b => Tìm trọng số lớn nhất của các cạnh nằm trên đường đi từ a đến b

Input

Input là một bộ gồm nhiều test. Dòng đầu của input là số test t ( t<=20 ). Tiếp sau đó là các test.   
Mỗi test bắt đầu bằng một dòng trống. Dòng tiếp theo ghi một số N ( N<=10000 ). N-1 dòng tiếp theo, mỗi dòng ghi 3 số a, b và c mô tả một cạnh của cây nối a với b và có trọng số là c. Thứ tự của các cạnh chính là thứ tự xuất hiện trong input. Tiếp theo là dãy các lệnh như mô tả ở trên(số lệnh không quá 50000). Cuối mỗi test ghi một từ "DONE".   
Dữ liệu vào luôn đảm bảo trọng số của các cạnh ở mỗi thời điểm có giá trị tuyệt đối không vượt quá 10000000.

Output

Với mỗi lệnh "QUERY", in ra kết quả tìm được. Nếu a = b thì ghi ra 0.

Example

**Input:**

1

3

1 2 1

2 3 2

QUERY 1 2

CHANGE 1 3

QUERY 1 2

DONE

**Output:**

1

3

* **Người up:** beo\_map
* **Điểm:** 1.00
* **Ngôn ngữ cho phép:**
* **Nguồn bài:** Ðược add lên bởi Khúc Anh Tuấn

Xâu đối xứng ( PALINX )

**Giới hạn**

* **Thời gian:** 0.496s
* **Bộ nhớ:** 1536MB
* **Mã nguồn:** 50000 bytes

Một xâu được gọi là đối xứng nếu đọc từ trái qua phải và đọc từ phải qua trái đều giống nhau.   
Ví dụ xâu "aba", "abba" là xâu đối xứng; còn xâu "abc", "abca" thì không.   
Bạn được cho N xâu, như vậy sẽ có NxN cặp xâu. Bạn hãy đếm xem trong NxN cặp xâu này, có bao nhiêu cặp mà khi nối xâu thứ hai vào sau xâu thứ nhất sẽ cho ra một xâu đối xứng.

Input

Dòng đầu ghi một số N. N dòng sau mỗi dòng mô tả một xâu, bắt đầu là độ dài của xâu, sau đó là một dấu cách và tiếp theo là nội dung của xâu.(Xâu chỉ gồm các chữ cái latin thường và có độ dài nguyên dương)   
Dữ liệu vào luôn đảm bảo tổng độ dài các xâu không quá 1000000.

Output

Ghi ra một số duy nhất là số cặp xâu tìm được.

Example

Input:

3

1 a

2 ab

2 ba

Output:

5

* **Người up:** beo\_map
* **Điểm:** 0.79
* **Ngôn ngữ cho phép:**
* **Nguồn bài:** Ðược add lên bởi Khúc Anh Tuấn

Chỉnh đồng hồ ( CLOCK )

**Giới hạn**

* **Thời gian:** 0.147s
* **Bộ nhớ:** 1536MB
* **Mã nguồn:** 50000 bytes

 9 chữ số trong hình 1 là vị trí của 9 đồng hồ, trong đó mỗi đồng hồ có 1 trong 4 vị trí: Bắc (12 giờ) , Đông ( 3 giờ ) , Nam ( 6 giờ ) và Tây ( 9 giờ) . ( Các hướng được đánh số tương ứng từ 0 -> 3 ):   
   
Có 9 cách khác nhau để quay các đồng hồ, mỗi cách được gọi là một dịch chuyển (Move). Mỗi dịch chuyển được chọn bằng một số từ 1 đến 9. Số đó sẽ quay các đồng hồ được đánh số 1 một góc 90 độ theo chiều kim đồng hồ. Các đồng hồ có đánh số 0 sẽ không bị tác động gì. 9 di chuyển được minh họa trong hình 2.  Hãy viết chương trình tính xem cần ít nhất bao nhiêu lần di chuyển để tất cả các đồng hồ đều chỉ 12h .

Input

Gồm 3 dòng , mỗi dòng gồm 3 chữ số cho biết hướng mà đồng hồ đang chỉ.

Output

Số lần dịch chuyển ít nhất.

Ví dụ

Input:

330

222

212

Output:

4

Giải thích test ví dụ : Thực hiện các phép biến đổi 5, 8, 4 và 9 .

* **Người up:** hard7771988
* **Điểm:** 0.15
* **Ngôn ngữ cho phép:**
* **Nguồn bài:** IOI 1994

Vòng số nguyên tố ( PCIRCLE )

**Giới hạn**

* **Thời gian:** 10.0s
* **Bộ nhớ:** 1536MB
* **Mã nguồn:** 50000 bytes

Một vòng tròn chứa 2\*n vòng tròn nhỏ (Xem hình vẽ). Các vòng tròn nhỏ được đánh số từ 1 đến 2\*n theo chiều kim đồng hồ. Cần điền các số tự nhiên từ 1 đến 2\*n mỗi số vào một vòng tròn nhỏ sao cho tổng của hai số trên hai vòng tròn nhỏ liên tiếp là số nguyên tố. Số điền ở vòng tròn nhỏ 1 luôn là số 1.

Input

Số nguyên dương n ( 1 < n < 10 ) .

Output

Dòng đầu tiên ghi ra số k là số cách tìm được.   
K dòng tiếp theo mỗi dòng ghi ra 1 cách điền các số vào các vòng tròn nhỏ. Cách điền nào có thứ tự từ điển nhỏ hơn thì xếp trước. Nếu K > 10000 thì chỉ cần ghi ra 10000 cách đầu tiên.

Ví dụ

Input:

4

Output:

4

1 2 3 8 5 6 7 4

1 2 5 8 3 4 7 6

1 4 7 6 5 8 3 2

1 6 7 4 3 8 5 2

* **Người up:** hard7771988
* **Điểm:** 0.08
* **Ngôn ngữ cho phép:**
* **Nguồn bài:** Mr Lê Minh Hoàng

Trò chơi đen trắng ( BWGAME )

**Giới hạn**

* **Thời gian:** 0.1s
* **Bộ nhớ:** 1536MB
* **Mã nguồn:** 50000 bytes

Có N ô vuông xếp cạnh nhau thành một dãy dài(N<=100000). Một số ô vuông đã được tô màu trắng, một số ô vuông đã được tô màu đen, còn lại là những ô chưa được tô màu. Không có 2 ô kề nhau nào cùng màu.   
Hai người cùng chơi một trò chơi, lượt chơi của 2 người xen kẽ nhau. Mỗi người chơi khi đến lượt mình sẽ tô màu vào ô chưa được tô với điều kiện 2 ô kề nhau không cùng màu. Ai đến lượt mình mà không chơi được nữa thì thua. Biết cả 2 người đều chơi tối ưu nhất có thể, hỏi ai là người thắng cuộc.

Input

Input gồm không quá 20 test, dòng đầu là số test. Sau đó là mô tả của các test.   
Mỗi test gồm 2 dòng, dòng đầu là số nguyên N, dòng sau ghi N ký tự với ý nghĩa : 0 - ô chưa được tô; 1 - ô đã tô màu trắng; 2 - ô đã tô màu đen.

Output

Ứng với mỗi test ghi một dòng duy nhất là 1 nếu người chơi đầu thắng, 2 nếu người chơi sau thắng.

Example

Input:

2

5

00100

4

1020

Output:

2

1

* **Người up:** beo\_map
* **Điểm:** 0.99
* **Ngôn ngữ cho phép:**
* **Nguồn bài:** Ðược add lên bởi Khúc Anh Tuấn

Trò chơi vòng số ( IVANA )

**Giới hạn**

* **Thời gian:** 0.009s
* **Bộ nhớ:** 1536MB
* **Mã nguồn:** 50000 bytes

Zvonko và Ivana cùng chơi trò chơi sau:

* Ban đầu, Zvonko đặt N số nguyên dương lên một vòng tròn.
* Mỗi người chơi chọn một số bên cạnh một số đã được chọn trước đó.
* Ivana đi trước, trong lượt đầu này Ivana được quyền chọn bất kỳ số nào.
* Trò chơi kết thúc khi tất cả các số đã được chọn. Ai chọn được nhiều số lẻ hơn sẽ thắng!

Bạn hãy giúp Ivana tìm những nước đi đầu tiên để sau đó cô có cơ hội thắng (biết rằng Zvonko luôn sử dụng chiến thuật tối ưu).

Input

* Dòng 1: n, số số trên vòng tròn. (1<=n<=100)
* Dòng 2: n số nguyên dương cách nhau bởi khoảng trắng. Các số trong phạm vi từ 1 đến 1000 và không có hai số nào bằng nhau.

Output

Gồm một số nguyên duy nhất, là số nước đi đầu tiên để Ivana có cơ hội thắng

Example

Input:

3

3 1 5

Output:

3

Input:

4

1 2 3 4

Output:

2

Input:

8

4 10 5 2 9 8 1 7

Output:

5

* **Người up:** paulmcvn
* **Điểm:** 0.38
* **Ngôn ngữ cho phép:**
* **Nguồn bài:** Croatian Open 2006

Tam giác vuông ( PRAVO )

**Giới hạn**

* **Thời gian:** 0.855s
* **Bộ nhớ:** 1536MB
* **Mã nguồn:** 50000 bytes

Cho n điểm trên mặt phẳng. Hỏi có bao nhiêu tam giác vuông được tạo thành.

Input

* Dòng đầu tiên chứa số nguyên dương n (3<=n<=1500), số điểm trên mặt phẳng
* Dòng thứ i trong n dòng tiếp theo, mỗi dòng chứa 2 số nguyên x i, y i, tọa độ của một điểm (-10 9<=x i, y i<= 10 9). Không có hai điểm nào có cùng tọa độ.

Output

Gồm một dòng duy nhất là số lượng tam giác vuông tìm được.

Example

Input:

3

4 2

2 1

1 3

Output:

1

Input:

4

5 0

2 6

8 6

5 7

Output:

0

Input:

5

-1 1

-1 0

0 0

1 0

1 1

Output:

7

* **Người up:** paulmcvn
* **Điểm:** 0.39
* **Ngôn ngữ cho phép:**
* **Nguồn bài:** Croatian Open 2007

Lát gạch ( LATGACH )

**Giới hạn**

* **Thời gian:** 0.697s
* **Bộ nhớ:** 1536MB
* **Mã nguồn:** 50000 bytes

Cho một hình chữ nhật kích thước 2xN (1<=N<=100). Hãy đếm số cách lát các viên gạch nhỏ kích thước 1x2 và 2x1 vào hình trên sao cho không có phần nào của các viên gạch nhỏ thừa ra ngoài, cũng không có vùng diện tích nào của hình chữ nhật không được lát.

Input

Gồm nhiều test, dòng đầu ghi số lượng test T ( T<=100 ).   
T dòng sau mỗi dòng ghi một số N.

Output

Ghi ra T dòng là số cách lát tương ứng.

Example

Input:

3

1

2

3

Output:

1

2

3

* **Người up:** beo\_map
* **Điểm:** 0.05
* **Ngôn ngữ cho phép:**
* **Nguồn bài:** Basic problem

Lát gạch 2 ( LATGACH2 )

**Giới hạn**

* **Thời gian:** 0.637s
* **Bộ nhớ:** 1536MB
* **Mã nguồn:** 7000 bytes

Trước tiên các bạn cần xem bài LATGACH ( level 1 ).   
Xét một hình vuông kích thước (n+2)x(n+2) (3<=n<=1000). Hình vuông này bị khuyết mất một diện tích là hình vuông kích thước (n-2)x(n-2) ở trung tâm ( Như vậy chỉ còn 4 đường biên với độ rộng là 2 ). Bạn vẫn được quyền sử dụng các viên gạch kích thước 1x2 và 2x1 để lát vừa kín hình trên. Hãy đếm số cách lát.

Input

Gồm nhiều dòng, mỗi dòng ghi 1 số n của test tương ứng.

Output

Với mỗi số n trong input, bạn cần đưa ra một số tương ứng ở output thể hiện số cách lát.

Example

Input:

3

50

Output:

196

401520777816387690468975409805843797460004

* **Người up:** beo\_map
* **Điểm:** 0.53
* **Ngôn ngữ cho phép:**
* **Nguồn bài:** Mở rộng của bài lát gạch

Cờ vua ( CHESS\_ )

**Giới hạn**

* **Thời gian:** 0.1s
* **Bộ nhớ:** 1536MB
* **Mã nguồn:** 7000 bytes

Cho 1 bàn cờ kích thước 3 dòng, N cột (N<=10^9). Các ô thuộc dòng trên cùng có một quân tốt đen ở mỗi ô, các ô ở dòng dưới cùng có một quân tốt trắng ở mỗi ô.   
Hai người chơi 1 trò chơi, quân trắng đi trước. Hai người thực hiện nước đi của mình luân phiên, ai đến lượt mình mà không thể đi được nữa ( do không còn nước đi hợp lệ ) sẽ thua. Tại mỗi nước đi, một người sẽ đi các quân tốt theo luật cờ vua, 2 luật quan trọng nhất là :   
\*\* Quân tốt chỉ có thể đi 1 ô dọc mỗi nước, và không được lùi.   
\*\* Quân tốt có thể ăn quân tốt khác màu theo 1 ô chéo.   
Tuy nhiên có 2 luật sau khác với luật cờ vua :   
\*\* Quân tốt không thể phong cấp ( thành hậu, mã, ...).   
\*\* Tại một nước đi, việc ăn tốt khác là bắt buộc ( nếu tồn tại nước ăn thì người chơi bắt buộc phải ăn tốt đối phương, nếu có nhiều lựa chọn ăn thì sự lựa chọn thuộc về người chơi).   
Biết cả 2 người chơi đều cố gắng chơi tốt nhất có thể. Hãy xác định ai là người chiến thắng.

Input

Gồm nhiều dòng, mỗi dòng ghi một số N.

Output

Với mỗi số N tương ứng ở input, in ra "White" nếu người chơi quân trắng thắng, in ra "Black" nếu người chơi quân đen thắng. ( Không in ra dấu nháy kép " )

Example

Input:

3

4

5

Output:

White

Black

White

* **Người up:** beo\_map
* **Điểm:** 0.77
* **Ngôn ngữ cho phép:**
* **Nguồn bài:** Một bài cơ bản về lý thuyết trò chơi

Rút gọn đoạn ( CUTSEG )

**Giới hạn**

* **Thời gian:** 0.429s
* **Bộ nhớ:** 1536MB
* **Mã nguồn:** 7000 bytes

Cho một dãy gồm N chữ số thuộc đoạn 0..9 (N<=200). Ở mỗi bước, ta có thể lấy ra từ dãy này một đoạn liên tiếp các chữ số giống nhau và nhận được một số tiền bằng bình phương độ dài của đoạn được lấy ra. Nếu sau khi lấy, dãy đã cho bị tách làm 2 dãy con, 2 dãy con này lập tức được sát nhập lại thành 1 ( giữ nguyên thứ tự ).   
Hãy tính số lượng tiền lớn nhất có thể thu được.

Input

Dòng đầu ghi số N. Dòng thứ hai ghi N chữ số thể hiện dãy.

Output

Ghi ra số lượng tiền lớn nhất có thể thu được.

Example

Input:

6

100011

Output:

18

* **Người up:** beo\_map
* **Điểm:** 0.58
* **Ngôn ngữ cho phép:**
* **Nguồn bài:** Một bài của đội tuyển Ðà Nẵng

Trò chơi với những đồng xu ( GWCOINS )

**Giới hạn**

* **Thời gian:** 0.1s
* **Bộ nhớ:** 1536MB
* **Mã nguồn:** 5000 bytes

Tuấn và Hiếu rất thích chơi xu lẻ. Một lần, Tuấn rủ Hiếu chơi một trò chơi. Ban đầu Tuấn kẻ một hàng gồm N ô vuông liên tiếp trên nền nhà, sau đó Tuấn lần lượt đặt các đồng xu vào một số ô ( không có ô nào có quá 1 đồng xu ). Hai người bắt đầu thực hiện nước đi luân phiên nhau, Tuấn đi trước. Ở mỗi nước đi, một người phải chọn 1 đồng xu bất kỳ và một ô còn trống phía bên phải đồng xu này. Đồng xu được chọn sẽ được đặt vào vị trí mới và mọi đồng xu nằm giữa vị trí cũ và mới của đồng xu đã chọn đều bị di chuyển sang trái một ô. Người nào đến lượt mà không thể thực hiện nước đi sẽ là người thua cuộc. Sau một thời gian chơi trò chơi này, Hiếu không thắng được ván nào. Hiếu bắt đầu nghi ngờ rằng, vị trí ban đầu của các đồng xu luôn đảm bảo cho Tuấn một thắng lợi. Bạn hãy kiểm chứng điều này.

Input

Gồm nhiều dòng, mỗi dòng là một xâu chỉ gồm ký tự 'C' và '.' biểu diễn trạng thái ban đầu của trò chơi. 'C' thể hiện ô vuông đó có một đồng xu và '.' thể hiện một ô vuông trống. Ký tự đầu tiên đến ký tự cuối cùng của xâu thể hiện các ô vuông từ trái sang phải của trò chơi. Mỗi dòng có không quá 500 ký tự.

Output

Với mỗi dòng của input, in ra một dòng tương ứng. "Tuan" thể hiện rằng Tuấn luôn là người chiến thắng bất chấp nỗ lực của Hiếu, "Hieu" trong trường hợp ngược lại. Nếu trạng thái trò chơi lúc ban đầu đảm bảo tính công bằng cho cả 2 người chơi, in ra "Cong bang". ( Các xâu in ra không có dấu " ).

Example

Input:

C.C.C

Output:

Tuan

Giải thích:

Gọi (a,b) là nước đi di chuyển đồng xu ở ô a đến ô trống b, các ô đánh số từ 1 đến N từ trái sang phải. Nếu bước đầu tiên, Tuấn đi (3,4), Hiếu sẽ đi (1,3) và dành chiến thắng. Nếu Tuấn đi (1,4), đồng xu ở ô 3 sẽ bị dịch sang trái 1 ô, Hiếu sẽ đi tiếp (2,3) và chiến thắng. Nếu Tuấn đi (1,2), Hiếu có thể đi (3,4) hoặc (2,4), nhưng Tuấn sẽ dành chiến thắng mà không phụ thuộc vào sự lựa chọn của Hiếu.

* **Người up:** beo\_map
* **Điểm:** 1.13
* **Ngôn ngữ cho phép:**
* **Nguồn bài:** Ðược gửi lên bởi Khúc Anh Tuấn

**Trang 3**

Đếm các hình chữ nhật ( CRECT )

**Giới hạn**

* **Thời gian:** 0.396s
* **Bộ nhớ:** 1536MB
* **Mã nguồn:** 5000 bytes

HùngĐM đang theo học một khóa học cơ bản về tiếng Đức. Vì mới bắt đầu, HùngĐM mới biết mặt 5 chữ cái là A, B, C, D, E. Ngày sinh nhật, HùngĐM được tặng một bảng hình chữ nhật có các chữ cái ghi ở các ô. Nhiệm vụ của HùngĐM sẽ là tìm các từ ẩn trong bảng này. Tuy nhiên, do vốn lười học ngoại ngữ, đam mê lập trình, HùngĐM lại nghĩ ra một trò chơi khác: đếm số hình chữ nhật con của bảng này có chứa đúng 3 chữ cái khác nhau( Vì HùngĐM không thích quá ít, cũng chẳng ưa quá nhiều ). Tuy nhiên, bài này không đơn giản để HùngĐM có thể giải được dễ dàng. Các bạn hãy giúp HùngĐM để HùngĐM có thể nhanh chóng tập trung vào việc học tiếng Đức.

Input

Dòng đầu ghi 2 số M, N (M, N <= 400). Bảng chữ của HùngĐM được chia làm M dòng, mỗi dòng gồm N ô vuông đơn vị. M dòng sau, mỗi dòng là một xâu độ dài N thể hiện một dòng của bảng chữ chỉ gồm các chữ cái A, B, C, D, E.

Output

Gồm một số duy nhất là số hình chữ nhật con tìm được.

Example

Input:

4 3

CED

CEB

CBC

DDA

Output:

12

* **Người up:** beo\_map
* **Điểm:** 0.33
* **Ngôn ngữ cho phép:**
* **Nguồn bài:** Ðược gửi lên bởi Khúc Anh Tuấn

Đua xe công thức 1 ( F1 )

**Giới hạn**

* **Thời gian:** 0.075s
* **Bộ nhớ:** 1536MB
* **Mã nguồn:** 50000 bytes

Các bạn chắc đều biết đến môn thể thao đua xe công thức 1 - môn thể thao của tốc độ. Tuy nhiên, thời gian gần đây, để tăng tính hấp dẫn của môn thể thao này, BTC quyết định thay đổi luật lệ chơi. Các tay đua sẽ đua xe trong một khu vực hình chữ nhật chia làm M x N ô nhỏ. Ở mỗi ô có một điểm số nhất định ( tất nhiên có thể là số âm, đó là các chướng ngại vật nguy hiểm ). Các tay đua xuất phát ở vị trí ô trái trên của bản đồ, có thể đi từ ô này sang ô khác kề cạnh nhưng không được phép rẽ trái, cũng không được phép đi vào ô đã từng đi qua. Tay đua có quyền quyết định kết thúc chặng đua ở bất cứ đâu. Sau khi kết thúc cuộc đua, ai là người có vận tốc nhanh nhất sẽ chiến thắng. Tuy nhiên, trong trường hợp 2 người có cùng vận tốc ( điều này rất hay xảy ra do thiết bị đo cũ kỹ, chỉ đo được với độ chính xác 100Km/h :D ), người nào đạt được nhiều điểm hơn sẽ chiến thắng. Điểm của từng tay đua sẽ bằng tổng điểm các ô tay đua đó đã đi qua. Bạn là một trong các vận động viên tham gia cuộc đua này, hãy tính toán xem đường đua nào sẽ đem lại cho bạn nhiều điểm nhất.

Input

Dòng đầu ghi 2 số M, N lần lượt là 2 kích thước của bản đồ ( M, N <= 20 ). M dòng sau mỗi dòng ghi N số là điểm số của ô tương ứng. Điểm số nằm trong khoảng từ -100 đến 100.

Output

Một số duy nhất là điểm số lớn nhất có thể đạt được.

Example

Input:

2 2

-33 37

15 -5

Output:

14

* **Người up:** beo\_map
* **Điểm:** 0.80
* **Ngôn ngữ cho phép:**
* **Nguồn bài:** (none)

Đua xe công thức 2 ( F2 )

**Giới hạn**

* **Thời gian:** 0.457s
* **Bộ nhớ:** 1536MB
* **Mã nguồn:** 50000 bytes

Đường đua xe công thức 2 năm nay nằm trong một khuôn viên hình chữ nhật chia làm M x N ô nhỏ. Tuy nhiên trong khuôn viên này có một số ô chướng ngại vật và không thể đi vào. Tay đua cần xuất phát từ 1 ô bất kỳ, đi qua tất cả các ô không có chướng ngại vật, mỗi ô đúng một lần rồi quay về điểm xuất phát. Tay đua chỉ có thể đi từ 1 ô sang các ô kề cạnh. Hãy đếm số đường đua khác nhau có thể.

Input

Dòng đầu ghi 2 số M, N ( M, N <= 12 ). M dòng sau, mỗi dòng ghi N ký tự. Ký tự '\*' thể hiện ô tương ứng có chướng ngại vật, và '.' nếu ngược lại.

Output

Gồm 1 số duy nhất ghi ra tổng số đường đua. Dữ liệu luôn đảm bảo kết quả nhỏ hơn 2^63-1.

Example

Input:

4 4

\*\*..

....

....

....

Output:

2

* **Người up:** beo\_map
* **Điểm:** 1.78
* **Ngôn ngữ cho phép:**
* **Nguồn bài:** acm.timus.ru

Rào Vườn ( RAOVUON )

**Giới hạn**

* **Thời gian:** 0.134s
* **Bộ nhớ:** 1536MB
* **Mã nguồn:** 10000 bytes

Bạn Minh Đức là chủ một vườn cây ăn quả lớn ở miền Nam. Nếu nhìn từ trên cao xuống, các gốc cây giống như các điểm trên mặt phẳng tọa độ là mặt đất. Đã nhiều năm rồi, bạn Minh Đức không được bội thu do nạn đạo tặc. Do vậy, năm ngoái bạn Minh Đức quyết tâm rào khu vườn của mình lại. Để làm được điều này, bạn Minh Đức chăng đường rào theo các gốc cây để tạo thành một đa giác bao kín vườn cây. Do tính keo kiệt và chi phí của đường rào là rất đắt, bạn Minh Đức đã tính toán chi li để đường rào có chu vi nhỏ nhất có thể. Chắc các bạn cũng biết đây là bài toán tin cơ bản : tìm bao lồi nhỏ nhất của một tập điểm. Tuy nhiên, năm nay bạn Minh Đức cũng không thu hoạch được thêm nhiều. Lý do là có một số cây ở đường biên của hàng rào vẫn không thoát khỏi bàn tay của đạo tặc. Năm nay bạn Minh Đức quyết xây dựng lại hàng rào để cho không còn cây nào nằm ở đường biên nữa. Để làm được điều này, thay vì chăng đường rào theo các gốc cây, bạn Minh Đức sẽ chăng đường rào theo các cột sắt có sẵn trong vườn. Vị trí của các cây và cột sắt đã rõ, nhưng xây dựng làm sao để hàng rào có chu vi nhỏ nhất vẫn là vấn đề nan giải. Bạn hãy giúp bạn Minh Đức giải quyết bài toán khó trên và cùng chia sẻ một vụ mùa bội thu.

Input

Dòng đầu là số N ( N <= 100 ). Là số cây trong vườn.   
N dòng sau, mỗi dòng ghi 2 số là tọa độ của một cây trong vườn.   
Dòng tiếp theo là số M ( M <= 100 ). Là số cột sắt trong vườn.   
M dòng sau, mỗi dòng ghi 2 số là tọa độ của một cột sắt.   
Các tọa độ đều là số nguyên trong khoảng -10000..10000.

Output

In ra một số duy nhất là độ dài nhỏ nhất của hàng rào với đúng 2 chữ số sau dấu chấm thập phân ( có làm tròn ). Dữ liệu luôn đảm bảo có ít nhất 1 cách xây hàng rào thỏa mãn.

Example

Input:

1

0 2

3

-2 0

2 0

0 4

Output:

12.94

* **Người up:** beo\_map
* **Điểm:** 1.14
* **Ngôn ngữ cho phép:**
* **Nguồn bài:** (none)

Tổng các ước chung lớn nhất ( GCDSUM )

**Giới hạn**

* **Thời gian:** 0.14s
* **Bộ nhớ:** 1536MB
* **Mã nguồn:** 10000 bytes

Một lần, ktuan được thầy giáo cho bài tập về nhà, yêu cầu tính tổng tất cả các ước chung lớn nhất của các cặp số (i, j) thỏa mãn : 1<=i< j<=N ( N là một số tự nhiên cho trước ). Rất nhanh chóng, ktuan đã cho ra một đoạn code như sau:   
  
for i:=1 to N-1 do for j:=i+1 to N do sum := sum + gcd( i, j);   
  
với gcd là hàm tính ước chung lớn nhất của 2 số, sum chính là kết quả cuối cùng.   
Thầy giáo yêu cầu ktuan dùng chương trình trên để tính kết quả với N = 1000000. Tuy nhiên, chương trình trên chạy quá lâu. Để khắc phục vấn đề, ktuan đã viết lại đoạn mã đó bằng C++ ( với hi vọng C++ sẽ chạy nhanh hơn pascal nhiều ) :   
  
for(int i=1;i< N;++i) for(int j=i+1;j<=N;++j) sum += gcd(i,j);   
  
Thật không may, đoạn chương trình trên vẫn không giải quyết được vấn đề, bạn hãy giúp ktuan giải đáp yêu cầu của thầy giáo.   
  
Lưu ý: bài này có thể giải bằng phương pháp Quy hoạch động và các kiến thức sơ đẳng trong toán học, không cần sử dụng những kiến thức toán học phức tạp không nằm trong phạm vi chương trình phổ thông.

Input

Gồm nhiều dòng, mỗi dòng là một số N ( 1<=N<=10^6) ứng với một test. Dữ liệu vào sẽ kết thúc sau khi gặp N=0 ( bạn không cần thực hiện test này ).

Output

Với mỗi giá trị của N, in ra một dòng là giá trị của sum sau khi thực hiện đoạn mã trên.

Example

Input:

4

0

Output:

7

* **Người up:** beo\_map
* **Điểm:** 0.56
* **Ngôn ngữ cho phép:**
* **Nguồn bài:** ACM World Final Warm up 1 - 2008

Số phụ thuộc ( SUMS )

**Giới hạn**

* **Thời gian:** 0.53s
* **Bộ nhớ:** 1536MB
* **Mã nguồn:** 50000 bytes

Cho tập số nguyên A gồm n phần tử, A={a1, a2,..., an}. Số k được gọi là phụ thuộc vào tập A, nếu k được tạo thành bằng cách cộng các phần tử của tập A(mỗi phần tử có thể cộng nhiều lần).

Ví dụ cho A={2,5,7}. Các số như 2, 4(2+2), 12(5+7 hoặc 2+2+2+2+2) được gọi là phụ thuộc vào tập A. Số 0 cũng gọi là phụ thuộc vào tập A.

Yêu cầu:

Cho một dãy B, hãy kiểm tra xem bi có phải là số phụ thuộc vào tập A hay không .

Dữ liệu:

* Dòng đầu tiên chứa số nguyên n (1 ≤ n ≤ 5000).
* N dòng tiếp theo chứa các phân tử của tập A, a1 < a2 < ... < an (1 ≤ ai ≤ 50000 ).
* Dòng thứ N+2 chứa số nguyên m (1 ≤ m ≤ 10000 ).
* M dòng tiếp theo chứa dãy số nguyên b1, b2, ..., bm (0 ≤ bi ≤ 1000000000 ).

Kết quả:

Gồm m dòng, dòng thứ i ghi ra TAK nếu bi là số phụ thuôc vào tập A và NIE nếu không phải là số phụ thuộc.

Ví dụ:

Dữ liệu :

3

2

5

7

6

0

1

4

12

3

2

Kết quả :

TAK

NIE

TAK

TAK

NIE

TAK

* **Người up:** aukcwe
* **Điểm:** 0.32
* **Ngôn ngữ cho phép:**
* **Nguồn bài:** Polish Olympiad in Infomatics 2003

Closest distance ( GANNHAT )

**Giới hạn**

* **Thời gian:** 0.35s
* **Bộ nhớ:** 1536MB
* **Mã nguồn:** 50000 bytes

Khoảng cách Manhattan giữa 2 điểm A(x 1,y 1) và B(x 2,y 2) được định nghĩa như sau:

D(A,B) = |x 1- x 2| + |y 1- y 2|

Cho N điểm A 1, A 2, ..., A Ntrên mặt phẳng, với mỗi điểm A ita cần tìm D(A i, A j) nhỏ nhất với j ≠ i.

Dữ liệu

* Dòng đầu ghi số nguyên dương N (1 ≤ N ≤ 200000).
* N dòng sau dòng thứ i ghi 2 số x và y là tọa độ của điểm thứ i. (0 ≤ x , y ≤ 10 7)

Kết quả

* Gồm N dòng, dòng thứ i ghi khoảng cách nhỏ nhất cần tìm đối với điểm thứ i.

Ví dụ

Dữ liệu:

4

0 0

0 1

1 0

1 1

Kết quả:

1

1

1

1

* **Người up:** aukcwe
* **Điểm:** 0.90
* **Ngôn ngữ cho phép:**
* **Nguồn bài:** Mr Taek

Phân nhóm ( GROUP )

**Giới hạn**

* **Thời gian:** 0.661s
* **Bộ nhớ:** 1536MB
* **Mã nguồn:** 50000 bytes

Cho n<=300000 cặp số (x,y) (1<=x, y<=1000000). Ta có thể nhóm một vài cặp số lại thành một nhóm. Giả sử một nhóm gồm các cặp số thứ a1, a2, ..., am thì chi phí cho nhóm này sẽ là max(x a1, x a2, ..., xam) \* max(y a1, y a2, ..., y am).

Yêu cầu: tìm cách phân nhóm có tổng chi phí bé nhất.

Dữ liệu

* Dòng đầu tiên là số nguyên dương N.
* N dòng tiếp theo dòng thứ i ghi 2 số xi và yi.

Kết quả

* Gồm 1 số duy nhất là kết quả tìm được.

Ví dụ

Dữ liệu:

4

100 1

15 15

20 5

1 100

Kết quả:

500

Giải thích:

Có 3 nhóm lần lượt là (1), (2,3) và (4).

* **Người up:** aukcwe
* **Điểm:** 0.59
* **Ngôn ngữ cho phép:**
* **Nguồn bài:** USACO MAR08

Số hiệu hoán vị ( SHHV )

**Giới hạn**

* **Thời gian:** 0.1s
* **Bộ nhớ:** 1536MB
* **Mã nguồn:** 50000 bytes

Xét tất cả các hoán vị của dãy số tự nhiên (1, 2,..., n) (1 <= n <= 12)

Giả sử rằng các hoán vị được sắp xếp theo thứ tự từ điển.

Yêu cầu:

1: Cho trước 1 hoán vị. Tìm số hiệu của hoán vị đó trong dãy đã sắp xếp

2: Cho trước số hiệu của 1 hoán vị trong dãy hoán vị đã sắp xếp. Tìm hoán vị đó

Input

Dòng 1: Chứa n số a1, a2, …, an ( dãy hoán vị n phần tử )

Dòng 2: Chứa số p ( số hiệu của hoán vị trong dãy hoán vị n phần tử )

Output

Dòng 1: Ghi số q ( số hiệu của dãy hoán vị a i)

Dòng 2: Ghi n số b 1, b 2, …, b n( dãy hoán vị có số hiệu p )

Example

Input:

2 1 3

4

Output:

3

2 3 1

* **Người up:** cun
* **Điểm:** 0.09
* **Ngôn ngữ cho phép:**
* **Nguồn bài:** IOICAMP

Hình vuông 0 1 ( QBSQUARE )

**Giới hạn**

* **Thời gian:** 0.137s
* **Bộ nhớ:** 1536MB
* **Mã nguồn:** 50000 bytes

Cho một bảng kích thước MxN, được chia thành lưới ô vuông đơn vị M dòng N cột ( 1 <= M, N <= 1000 )

Trên các ô của bảng ghi số 0 hoặc 1. Các dòng của bảng được đánh số 1, 2... M theo thứ tự từ trên xuống dưới và các cột của bảng được đánh số 1, 2..., N theo thứ tự từ trái qua phải

Yêu cầu:

Hãy tìm một hình vuông gồm các ô của bảng thoả mãn các điều kiện sau:

1 - Hình vuông là đồng nhất: tức là các ô thuộc hình vuông đó phải ghi các số giống nhau (0 hoặc 1)

2 - Cạnh hình vuông song song với cạnh bảng.

3 - Kích thước hình vuông là lớn nhất có thể

Input

Dòng 1: Ghi hai số m, n

M dòng tiếp theo, dòng thứ i ghi N số mà số thứ j là số ghi trên ô (i, j) của bảng

Output

Gồm 1 dòng duy nhất ghi kích thước cạnh của hình vuông tìm được

Example

Input:

11 13

0 0 0 0 0 1 0 0 0 0 0 0 0

0 0 0 0 1 1 1 0 0 0 0 0 0

0 0 1 1 1 1 1 1 1 0 0 0 0

0 0 1 1 1 1 1 1 1 0 0 0 0

0 1 1 1 1 1 1 1 1 1 0 0 0

1 1 1 1 1 1 1 1 1 1 1 0 0

0 1 1 1 1 1 1 1 1 1 0 0 0

0 0 1 1 1 1 1 1 1 0 0 0 0

0 0 1 1 1 1 1 1 1 0 0 0 0

0 0 0 0 1 1 1 0 0 0 0 1 1

0 0 0 0 0 1 0 0 0 0 0 1 1

Output:

7

* **Người up:** cun
* **Điểm:** 0.08
* **Ngôn ngữ cho phép:**
* **Nguồn bài:** Ðược add lên bởi Võ Khánh Trung

IOI2008 ( IOI2008 )

**Giới hạn**

* **Thời gian:** 0.312s
* **Bộ nhớ:** 1536MB
* **Mã nguồn:** 50000 bytes

IOI 2008 diễn ra trong n + 1 ngày, các bài toán của IOI được đánh số từ 1 tới n.(n+1) và được phân bố vào các ngày thi theo lịch sau (mỗi ngày thi có n bài toán):

Ngày 1: Các bài toán từ 1 tới n

Ngày 2: Các bài toán từ n + 1 tới 2n

...

Ngày i: Các bài toán từ (i - 1).n + 1 tới i.n

...

Ngày n+1: Các bài toán từ n.n + 1 tới n.(n+1)

Các bài thi có một trong k dạng, bài thứ j có dạng là r j(1 <= r j<= k)

Thể thức thi được thông báo cho mỗi đoàn như sau:

- Mỗi đoàn sẽ có n + 1 học sinh tham gia

- Hàng ngày, Ban tổ chức sẽ đưa một học sinh của đoàn đi tham quan thành phố, việc chọn học sinh nào cho đi tham quan là quyền của trưởng đoàn, nhưng phải đảm bảo điều kiện:

Cho đến khi IOI kết thúc, học sinh nào của đoàn cũng đã được đi tham quan thành phố. Như vậy mỗi ngày đoàn sẽ còn lại n học sinh tham gia thi, việc giao cho học sinh nào làm bài nào là quyền của phó đoàn nhưng mỗi học sinh chỉ được giao một bài và hai học sinh khác nhau sẽ phải nhận hai bài khác nhau.

Kết thúc IOI, điểm đồng đội của mỗi đoàn sẽ được tính bằng tổng điểm của tất cả các lời giải các bài toán đã cho.

Các thầy giáo trưởng, phó đoàn Việt Nam dự đoán rằng nếu học sinh thứ i của đoàn làm bài toán dạng j thì có thể thu được số điểm là c ij(c ij= 0 tương đương với lời dự đoán rằng học sinh thứ i không làm được bài toán dạng j).

Hỏi các thầy sẽ sắp xếp lịch thi đấu cho các học sinh như thế nào để theo dự đoán, đoàn Việt Nam sẽ thu được số điểm nhiều nhất có thể.

Input

Dòng 1: Chứa hai số n, k (1 <= n <= 100; 1 <= k <= 1000)

Dòng 2: Chứa n.(n+1) số, số thứ p là r p.

Các dòng tiếp, mỗi dòng chứa ba số nguyên dương i, j, p cho biết một điều dự đoán của các thầy: học sinh thứ i có thể làm được bài toán dạng j và đạt được số điểm là p (=c[i, j]). (1 <= p <= 100)

Output

Gồm 1 dòng duy nhất : Ghi điểm đồng đội mà theo dự đoán đoàn Việt Nam có thể đạt

Example

Input:

3 4

1 2 4 4 3 3 1 4 2 3 2 2

1 1 2

1 2 3

1 4 6

2 3 4

2 1 3

2 4 7

3 2 1

3 1 4

4 1 2

4 3 9

4 2 8

Output:

65

I hope and expect that you will have much success in IOI 2008 ^^

* **Người up:** cun
* **Điểm:** 1.95
* **Ngôn ngữ cho phép:**
* **Nguồn bài:** Lê Minh Hoàng

Hình chữ nhật 0 1 ( QBRECT )

**Giới hạn**

* **Thời gian:** 0.132s
* **Bộ nhớ:** 1536MB
* **Mã nguồn:** 50000 bytes

Cho một bảng kích thước MxN, được chia thành lưới ô vuông đơn vị M dòng N cột ( 1 <= M, N <= 1000 )

Trên các ô của bảng ghi số 0 hoặc 1. Các dòng của bảng được đánh số 1, 2... M theo thứ tự từ trên xuống dưới và các cột của bảng được đánh số 1, 2..., N theo thứ tự từ trái qua phải

Yêu cầu:

Hãy tìm một hình chữ nhật gồm các ô của bảng thoả mãn các điều kiện sau:

1 - Hình chữ nhật đó chỉ gồm các số 1

2 - Cạnh hình chữ nhật song song với cạnh bảng

3 - Diện tích hình chữ nhật là lớn nhất có thể

Input

Dòng 1: Ghi hai số M, N

M dòng tiếp theo, dòng thứ i ghi N số mà số thứ j là số ghi trên ô (i, j) của bảng

Output

Gồm 1 dòng duy nhất ghi diện tích của hình chữ nhật tìm được

Example

Input:

11 13

0 0 0 0 0 1 0 0 0 0 0 0 0

0 0 0 0 1 1 1 0 0 0 0 0 0

0 0 1 1 1 1 1 1 1 0 0 0 0

0 0 1 1 1 1 1 1 1 0 0 0 0

0 1 1 1 1 1 1 1 1 1 0 0 0

1 1 1 1 1 1 1 1 1 1 1 0 0

0 1 1 1 1 1 1 1 1 1 0 0 0

0 0 1 1 1 1 1 1 1 0 0 0 0

0 0 1 1 1 1 1 1 1 0 0 0 0

0 0 0 0 1 1 1 0 0 0 0 1 1

0 0 0 0 0 1 0 0 0 0 0 1 1

Output:

49

* **Người up:** cun
* **Điểm:** 0.08
* **Ngôn ngữ cho phép:**
* **Nguồn bài:** Ðược add lên bởi Võ Khánh Trung

Hành Tinh Đá ( NSP )

**Giới hạn**

* **Thời gian:** 0.154s
* **Bộ nhớ:** 1536MB
* **Mã nguồn:** 512 bytes

Nuga là một cô bé thông minh, rất thích phiêu lưu. Cô thường lang thang trong vũ trụ bằng con tàu siêu tốc AlphaX. Một hôm, con tàu của Nuga đi lạc vào một hành tinh lạ.

Hành tinh có dạng một hình vuông khổng lồ, có cạnh là N (năm ánh sáng). Do sức hút của hành tinh quá lớn, không một vật thể nào, kể cả ánh sáng có thể thoát ra ngoài được, chính vì thế mà hàng mấy tỉ năm nay, không ai phát hiện ra sự tồn tại của hành tinh này.

Trong lúc thăm dò, Nuga phát hiện ra hàng triệu “viên đá” hình cầu to bằng cả trái đất nằm rải rác khắp hành tinh, mỗi viên đá mang một năng lượng riêng. Khi quét bản đồ hành tinh lên máy tính, Nuga thấy một điều rất đặc biệt, đó là nếu chia hành tinh thành N hàng, N cột, đánh số từ 1 đến N theo chiều từ trái sang phải và từ trên xuống dưới thì mỗi viên đá nằm gọn trong một ô vuông.

Nhờ phân tích các số liệu, Nuga đã biết được bí quyết “mở cửa” Hành Tinh Đá để thoát ra ngoài là phải kết nối được sức mạnh của K viên đá thần kỳ. Tuy nhiên, Nuga phải nhanh chóng xác định vị trí của chúng.

Công việc này hóa ra lại phức tạp hơn Nuga tưởng tượng, bởi hành tinh quá rộng lớn. Nghĩ rằng sắp xếp lại các viên đá có thể giúp tìm kiếm nhanh hơn, Nuga đã sắp xếp các viên đá trên mỗi hàng theo thứ tự năng lượng tăng dần, rồi sau đó lại tiếp tục sắp xếp các viên đá trên mỗi cột theo thứ tự năng lượng tăng dần.

Đến đây thì vừa mệt vừa đói, Nuga chưa tìm được hướng đi tiếp theo. Bạn hãy giúp Nuga với!

Input

* Dòng đầu ghi số 2 nguyên dương: N là độ dài 1 cạnh của hành tinh, và K là số lượng viên đá thần kỳ.
* N dòng tiếp theo, mỗi dòng ghi N số nguyên. Các số nguyên khác nhau đôi một. Số nguyên ở dòng thứ I, cột thứ J thể hiện năng lượng của viên đá tại vị trí (I, J) trên bản đồ mà Nuga đã sắp xếp.
* K dòng sau, mỗi dòng ghi 1 số nguyên là năng lượng của 1 viên đá thần kỳ cần tìm. Dữ liệu đảm bảo tất cả các viên đá thần đều có trên hành tinh.

Output

* Gồm K dòng, mỗi dòng ghi 2 số nguyên I, J thể hiện vị trí của viên đá thần kỳ tương ứng với năng lượng đã cho.

Hạn chế

* Năng lượng của mỗi viên đá nằm trong khoảng [0, 2^31 - 1]
* N ≤ 1000, K ≤ 10000.

Ví dụ

Input:

2 2

1 2

3 4

3

1

Output:

2 1

1 1

Đề bài đã được giới hạn lại mã nguồn. Một số bài đã AC với source > 512B nên code lại với cách tốt hơn

* **Người up:** racer
* **Điểm:** 0.24
* **Ngôn ngữ cho phép:**
* **Nguồn bài:** ^Nuga Còi^

Khối tam chữ nhật II ( RECT3\_2 )

**Giới hạn**

* **Thời gian:** 0.467s
* **Bộ nhớ:** 1536MB
* **Mã nguồn:** 50000 bytes

Đề bài giống hệt bài [RECT3](http://vn.spoj.pl/problems/RECT3/).

Dữ liệu

* Dòng đầu tiên: hai số nguyên M, N (1 ≤ M, N ≤ 2000).
* M dòng tiếp theo, mỗi dòng gồm N số 0 hoặc 1 (viết liền nhau) mô tả bảng chữ nhật.

Kết quả

In ra diện tích của khối tam chữ nhật lớn nhất, hoặc 0 nếu không tìm được khối tam chữ nhật nào.

Ví dụ

Dữ liệu :

6 8

00000110

00111000

00110100

01111001

01111110

00111010

Kết quả :

15

**Giải thích**

Khối tam chữ nhật:

00000110

00**11**1000

00**11**0100

01**11**1001

0**111111**0

00**111**010

Có diện tích là 15

Khối tam chữ nhật khác có diện tích 14:

00000110

00**11**1000

00**11**0100

0**1111**001

0**1111**110

00**11**1010

* **Người up:** aukcwe
* **Điểm:** 1.70
* **Ngôn ngữ cho phép:**
* **Nguồn bài:** Mở rộng từ RECT3 của VM08 - add lên bởi Gothdn.

Triomino Game ( TRIOMINO )

**Giới hạn**

* **Thời gian:** 0.79s
* **Bộ nhớ:** 1536MB
* **Mã nguồn:** 50000 bytes

 X và Y đang chơi trò chơi như sau: Họ bắt đầu với một bảng ô vuông trống kích thước 2xN và luân phiên chơi. X đi trước. Trong một nước đi, một người chơi phải đặt một quân L-triomino (như hình bên phải) vào bảng, theo một trong 4 hướng, và không được đặt chèn lên các ô đã có triomio. Người nào không thể thực hiện nước đi nữa thì thua.

Cho kích thước của bảng, tính xem ai sẽ là người chiến thắng nếu giả sử cả X và Y đều chơi tối ưu.

Input

Input chứa nhiều test. Dòng đầu tiên ghi số nguyên T là số lượng test. (T<=50)

Mỗi test chứa số nguyên N trên 1 dòng, là kích thước của bảng (N<=800)

Output

Với mỗi test, in ra trên một dòng kí tự "X" hoặc "Y" thể hiện ai là người chiến thắng trò chơi đó.

Example

Input:

2

3

4

Output:

X

Y

* **Người up:** racer
* **Điểm:** 0.91
* **Ngôn ngữ cho phép:**
* **Nguồn bài:** Byte Code

Vần hoàn hảo ( NPR )

**Giới hạn**

* **Thời gian:** 3.168s
* **Bộ nhớ:** 1536MB
* **Mã nguồn:** 50000 bytes

Đây là một bài để các bạn luyện tập về Trie

Cho một danh sách từ L và một từ w . Nhiệm vụ của bạn là phải tìm một từ trong L tạo thành một "vần hoàn hảo" với w . Từ u này là duy nhất xác định bởi các thuộc tinh sau:

* Nó nằm trong L .
* Nó khác w .
* Phần hậu tố chung của chúng dài nhất có thể.
* u là từ có thứ tự từ điển nhỏ nhất thoả mãn các điều trên

Chú ý

Một tiền tố của một từ là một chuỗi có thể thu được bằng cách lặp lại việc xoá kí tự cuối cùng của từ. Tương tự, một hậu tố của một từ là một chuỗi mà có thể thu được bằng cách lặp lại việc xoá kí tự đầu tiên của từ.

Ví dụ với từ: different .

Từ này vừa là tiền tố, vừa là hậu tố của chính nó. một tiền tố dài nhất khác của nó differen , và một hậu tố dài nhất khác của nó là ifferent . Chuỗi rent cũng là một hậu tố khác nhưng ngắn hơn. Chuỗi eent và iffe đều không phải là tiền tố hay hậu tố của từ different .

Gọi u và v là 2 từ khác nhau. Ta nói rằng u có thứ tự từ điển nhỏ hơn v nếu hoặc u là một tiền tố của v , hoặc nếu i là vị trí đầu tiên mà chúng khác nhau, và kí tự thứ i của u đứng trước kí tự thứ i của v trong bảng chữ cái.

Ví dụ, dog nhỏ hơn dogs , từ này lại nhỏ hơn dragon (Vì o nhỏ hơn r ).

Dữ liệu

Có 2 phần. Phần thứ nhất chứa danh sách từ L , mỗi từ trên 1 dòng. Mỗi từ chỉ chứa các chữ cái thường tiếng Anh và không có 2 nào từ giống nhau.

Phần thứ nhất kết thúc bằng một dòng trống.

Tiếp theo là phần 2, với mỗi câu hỏi cho từ w trên một dòng.

Bạn có thể chắc chắn rằng trong cả 2 phần của dữ liệu vào, độ dài của mỗi từ không quá 30. Và số lượng từ trong mỗi phần không quá 250000.

Kết quả

Với mỗi câu hỏi, viết ra trên một dòng từ mà tạo thành vần hoàn hảo với nó. Kết quả phải viết bằng chữ cái thường.

Ví dụ

Dữ liệu

perfect

rhyme

crime

time

crime

rhyme

Kết quả

time

crime

Trong câu hỏi thứ 2, có 2 từ có cùng độ dài hậu tố với rhyme (là crime và time), từ có thứ tự từ điển nhỏ hơn đã được chọn.

Cảnh báo: File dữ liệu và kết quả rất lớn, cẩn thận với một số ngôn ngữ

* **Người up:** racer
* **Điểm:** 0.54
* **Ngôn ngữ cho phép:**
* **Nguồn bài:** IPSC

Tập hợp động ( CPPSET )

**Giới hạn**

* **Thời gian:** 0.219s
* **Bộ nhớ:** 1536MB
* **Mã nguồn:** 50000 bytes

Cho một tập hợp S các số nguyên, bạn hãy lập trình thực hiện các thao tác sau:

* ADD x: thêm số x vào tập S
* DELETE x: xóa số x khỏi tập S
* MININUM: tìm số nhỏ nhất trong tập S
* MAXIMUM: tìm số lớn nhất trong tập S
* SUCC x: tìm số nhỏ nhất lớn hơn x trong tập S
* SUCC\_2 x: tìm số nhỏ nhất và không nhỏ hơn x trong tập S
* PRED x: tìm số lớn nhất nhỏ hơn x trong tập S
* PRED\_2 x: tìm số lớn nhất không vượt quá x trong tập S

Ghi chú: Đối với thao tác DELETE, giữ nguyên tập S nếu x không có trong tập S. Đối với các thao tác MINIMUM, MAXIMUM, SUCC, SUCC\_2, PRED và PRED\_2, in ra 'empty' nếu tập S rỗng. Đối với các thao tác SUCC, SUCC\_2, PRED và PRED\_2, in ra 'no' nếu không tìm được số thỏa mãn.

Các thao tác ADD, DELETE, MINIMUM, MAXIMUM, SUCC, SUCC\_2, PRED, PRED\_2 lần lượt được mã hóa bởi các chỉ số 1 2 3 4 5 6 7 8.

Dữ liệu

Gồm nhiều dòng, mỗi dòng bắt đầu bằng một số từ 0 đến 8 cho biết chỉ số thao tác cần thực hiện. Số 0 báo hiệu kết thúc dữ liệu nhập. Đối với các thao tác 1, 2, 5, 6, 7, 8, số tiếp theo trên dòng là số nguyên x (|x| ≤ 10 9) cho biết tham số của thao tác. Biết số thao tác cần thực hiện không vượt quá 300000.

Kết quả

Đối với mỗi thao tác loại 3, 4, 5, 6, 7, 8 in ra một dòng là kết quả của thao tác.

Ví dụ

Dữ liệu

4

1 10

1 5

3

1 7

4

2 5

3

5 10

6 10

7 10

8 10

2 10

7 100

0

Kết quả

empty

5

10

7

no

10

7

10

7

Gợi ý

Bạn hãy dùng cấu trúc set trong thư viện STL của C++. Bài này bạn chỉ nộp được bằng C++ mà thôi.

* **Người up:** paulmcvn
* **Điểm:** 0.33
* **Ngôn ngữ cho phép:**
* **Nguồn bài:** © VNOI

Binary multiplication ( MUL2COM )

**Giới hạn**

* **Thời gian:** 0.204s
* **Bộ nhớ:** 1536MB
* **Mã nguồn:** 50000 bytes

Đây là một số kiến thức về biểu diễn số trên máy tính trước khi các bạn giải bài toán này. Với n bit, máy tính có thể biểu diễn được 2 nsố trong phạm vi -2 n-1..(2 n-1-1). Máy tính sử dụng các số bù 2 để biểu diễn các số âm. Trong cách biểu diễn số bù 2 có n bit thì:

* Số x ≥ 0 được biểu diễn bằng chính số x.
* Số x < 0 được biểu diễn bằng số 2 n- x.

Dưới đây là bảng biểu diễn số bù 2 với 3 bit: Biễu diễn số bù 2 Giá trị 000 0 001 1 010 2 011 3 100 -4 101 -3 110 -2 111 -1

Ưu điểm của số bù 2 là phép cộng có thể thực hiện hoàn toàn như các số không dấu. Với x > 0, x + (-x) = 2 n= 0 vì các số chỉ được biểu diễn bởi n bit. Ví dụ, xét phép cộng (-2) + 3 trong biểu diễn số bù hai 3 bit:

110

+011

---

001

Kết quả bằng 1.

Với x ≥ 0, biểu diễn số bù 2 của -x sẽ thu được bằng cách đảo toàn bộ bit của x và cộng thêm 1 đơn vị. Nói cách khác -x = (NOT x) + 1. Để chỉ ra biểu thức này đúng, ta nhận xét rằng NOT x = 2 n-1-x, do đó (NOT x) + 1 = 2 n-x chính là biểu diễn của số -x. Với x < 0, ta cũng có -x = (NOT x) + 1.

Trong bài toán này, bạn cần thực hiện phép nhân hai số bù hai có n bit. Kết quả trả về cũng là một số bù hai n bit. Bạn hãy thông báo lỗi nếu kết quả vượt quả phạm vi biểu diễn.

Dữ liệu

Gồm nhiều bộ test, mỗi bộ test có dạng như sau:

Dòng đầu tiên chứa số n (0 ≤ n ≤ 1024) là số bit. n=0 cho biết kết thúc dữ liệu nhập. Nếu n>0, hai dòng tiếp theo, mỗi dòng chứa một số bù hai có n bit.

Có không quá 40 bộ test.

Kết quả

Gồm nhiều dòng, mỗi dòng tương ứng với một test chứa một số bù hai với số bit tương ứng là kết quả của phép nhân hoặc chứa chuỗi 'overflow' nếu kết quả vượt qua phạm vi biểu diễn.

Ví dụ

Dữ liệu

3

110

011

4

0011

1110

0

Kết quả

overflow

1010

* **Người up:** paulmcvn
* **Điểm:** 1.14
* **Ngôn ngữ cho phép:**
* **Nguồn bài:** © VNOI

Ai là sếp ( BOSS )

**Giới hạn**

* **Thời gian:** 0.23s
* **Bộ nhớ:** 1536MB
* **Mã nguồn:** 50000 bytes

Ở một công ty nọ có một quy định khá lạ lùng đó là Sếp thì không được thấp hơn nhân viên! Sếp trực tiếp của một nhân viên A được định nghĩa là nhân viên có chiều cao không thấp hơn A và có mức lương nhỏ nhất nhưng vẫn cao hơn A. Quan hệ lính được định nghĩa đệ quy như sau, nếu A là sếp của B thì B và các lính của B đều là lính của A.

Biết công ty này có N nhân viên và một số truy vấn đến các nhân viên. Với mỗi truy vấn, bạn hay tìm sếp trực tiếp và số lượng lính của nhân viên đó.

Dữ liệu

Dòng đầu tiên chứa số lượng test. Mỗi test có cấu trúc như sau:

* Dòng 1 chứa hai số N và Q là số nhân viên và số truy vấn (1 ≤ N ≤ 30000, 1 ≤ Q ≤ 200)
* N dòng tiếp theo mỗi dòng chứa thông tin về một nhân viên gồm 3 số nguyên: số hiệu của nhân viên (luôn có 6 chữ số, chữ số đầu tiên khác 0), lương và chiều cao. Lương không vượt quá 10000000. Chiều cao trong phạm vi 1000000 đến 2500000.
* Q dòng tiếp theo mỗi dòng chứa số hiệu của một nhân viên cần truy vấn.

Kết quả

Với mỗi truy vấn in ra hai số nguyên là số hiệu của sếp trực tiếp và số lính. Nếu không có sếp thì in ra số 0.

Ví dụ

Dữ liệu

2

3 3

123456 14323 1700000

123458 41412 1900000

123457 15221 1800000

123456

123458

123457

4 4

200002 12234 1832001

200003 15002 1745201

200004 18745 1883410

200001 24834 1921313

200004

200002

200003

200001

Kết quả

123457 0

0 2

123458 1

200001 2

200004 0

200004 0

0 3

* **Người up:** paulmcvn
* **Điểm:** 0.46
* **Ngôn ngữ cho phép:**
* **Nguồn bài:** NWERC 2003

34 đồng xu ( COIN34 )

**Giới hạn**

* **Thời gian:** 0.34s
* **Bộ nhớ:** 1536MB
* **Mã nguồn:** 50000 bytes

Bạn có 34 đồng xu có giá trị như sau:

xu(1) có giá trị 2

xu(2) có giá trị 3

xu(3) có giá trị 5

for n = 4 to 34

xu(n) có giá trị (xu(n-1) + xu(n-2) + xu(n-3))

Bạn hãy dùng nhiều đồng xu nhất để mua một món hàng có giá là X!

Dữ liệu

Dòng đầu tiên là số test (không quá 1000). Mỗi dòng tiếp theo chứa một số nguyên X (1 ≤ X ≤ 2000000000).

Kết quả

Với mỗi test, in ra "Case #" + số hiệu test + ": " + số lượng lớn nhất đồng xu cần dùng. Nếu không có cách nào để đạt giá trị X thì in ra -1.

Ví dụ

Dữ liệu

4

1

5

8

9

Kết quả

Case #1: -1

Case #2: 2

Case #3: 2

Case #4: -1

* **Người up:** paulmcvn
* **Điểm:** 0.15
* **Ngôn ngữ cho phép:**
* **Nguồn bài:** NUS ACM Training

Lũy thừa ( POWER )

**Giới hạn**

* **Thời gian:** 0.176s
* **Bộ nhớ:** 1536MB
* **Mã nguồn:** 50000 bytes

Cho l, r sao cho 2 ≤ l ≤ r ≤ 10 12. Tìm k nguyên dương lớn nhất sao cho tồn tại số nguyên dương x để l ≤ x k≤ r.

Dữ liệu

Dòng đầu tiên là số lượng test. Mỗi dòng tiếp theo chứa hai số nguyên l, r biểu thị một test.

Kết quả

Đối với mỗi test in ra "Case #" + số hiệu test + ": " + số k lớn nhất tìm được.

Ví dụ

Dữ liệu

4

5 20

10 12

2 100

1000000000000 1000000000000

Kết quả

Case #1: 4

Case #2: 1

Case #3: 6

Case #4: 12

* **Người up:** paulmcvn
* **Điểm:** 0.15
* **Ngôn ngữ cho phép:**
* **Nguồn bài:** NUS ACM Training

Phần tử trung vị ( MEDIAN )

**Giới hạn**

* **Thời gian:** 0.962s
* **Bộ nhớ:** 1536MB
* **Mã nguồn:** 50000 bytes

Cho một dãy số a1, a2, ..., an được sinh ngẫu nhiên như sau:

* a1=seed
* a i=(a i-1\*mul+add)%65536

Với mul, add, seed là các số cho trước.

Cho một số k ≤ n. Dãy đã cho có n-k+1 dãy con độ dài k. Hãy tính tổng tất cả các phần tử trung vị (phần tử nhỏ thứ (k+1)/2) của n-k+1 dãy con này).

Dữ liệu

Dòng đầu tiên chứa số test (không quá 30). Mỗi dòng tiếp theo chứa 5 số nguyên seed, mul, add, N, K. (0 ≤ seed, mul, add ≤ 65535, 1 ≤ N ≤ 250000, 1 ≤ K ≤ 5000, K ≤ N)

Kết quả

Với mỗi test in ra số hiệu test (theo mẫu) cùng với tổng các trung vị tìm được.

Ví dụ

Dữ liệu

5

3 1 1 10 3

10 0 13 5 2

4123 2341 1231 7 3

47 5621 1 125000 1700

32321 46543 32552 17 17

Kết quả

Case #1: 60

Case #2: 49

Case #3: 102186

Case #4: 4040137193

Case #5: 25569

Giải thích

Với test 1, dãy sinh ra là 3, 4, 5, 6, 7, 8, 9, 10, 11, and 12.

Các dãy con là (3, 4, 5), ..., (10, 11, 12).

Các trung vị là 4, 5, ..., 11.

* **Người up:** paulmcvn
* **Điểm:** 0.32
* **Ngôn ngữ cho phép:**
* **Nguồn bài:** NUS ACM Training

Số hiệu tổ hợp ( SHTH )

**Giới hạn**

* **Thời gian:** 0.1s
* **Bộ nhớ:** 1536MB
* **Mã nguồn:** 50000 bytes

Cho tập hợp A gồm N phần tử. Mỗi tập con gồm K (1 <= K <= N) phần tử của A được gọi là một tổ hợp chập K của N phần tử đã cho

Bài toán đặt ra là:

- Cho số hiệu của một tổ hợp chập K của N số nguyên dương đầu tiên, hãy tìm tổ hợp chập đó.

- Cho tổ hợp chập K của N số nguyên dương đầu tiên, hãy tính số hiệu của tổ hợp chập đó.

Input

Gồm 2 dòng có dạng như sau:

Dòng 1: Ghi 2 số nguyên N, K ( 3 <= N <= 300 )

Dòng 2: Ghi số nguyên S

Dòng 3: Gồm K số nguyên B 1, B 2, ... B K( B 1< B 2< ... < B K)

Output

Dòng 1: Ghi ra dãy số A 1, A 2, ... A Klà tổ hợp chập K của N số nguyên dương đầu tiên có số hiệu S. Các số viết theo thứ tự tăng dần.

Dòng 2: Ghi số hiệu của tổ hợp chập K: B 1, B 2, ... B K.

Example

Input:

3 2

2

2 3

Output:

1 3

3

* **Người up:** cun
* **Điểm:** 0.26
* **Ngôn ngữ cho phép:**
* **Nguồn bài:** Ðược add lên bởi Võ Khánh Trung

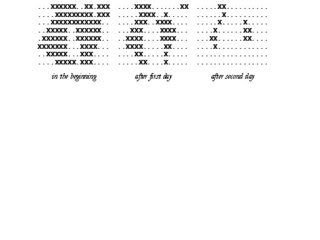
Hồ Thiên Nga ( LABUDOVI )

**Giới hạn**

* **Thời gian:** 0.254s
* **Bộ nhớ:** 1536MB
* **Mã nguồn:** 50000 bytes

Hai con thiên nga đang ở trong một cái hồ lớn, nhưng chúng lại đang bị chia cắt bởi băng đóng trong hồ nước. Hồ nước có dạng hình chữ nhật được chia thành R dòng C cột. Một số ô trong hồ bị băng đóng. Mùa xuân tới dần, băng trong hồ tan dần – mỗi ngày băng ở tất cả những ô tiếp xúc với nước đang ấm dần trong hồ (tức là kề cạnh một ô không bị đóng băng) sẽ tan ra.

Ví Dụ



Thiên nga có thể di chuyển tự do ở những ô chứa nước nhưng không thể đi qua những ô bị đóng băng. Bạn hãy tính xem sau bao nhiêu ngày thì đôi thiên nga của chúng ta có thể gặp nhau

Dữ liệu vào

* Dòng đầu tiên chứa 2 số R và C, 1 ≤ R, C ≤ 1500.
* Mỗi dòng trong R dòng tiếp theo chứa C kí tự mô tả hồ nước tại thời điểm hiện tại: '.' (dot) thể hiện 1 ô chứa nước, 'X' thể hiện 1 ô bị đóng băng, và 'L' thể hiện ô có thiên nga. Có chính xác 2 ô chữ L.

Dữ liệu ra

* Một dòng duy nhất chứa số ngày đôi thiên nga có thể gặp nhau.

Input:

10 2

.L

..

XX

XX

XX

XX

XX

XX

..

.L

Output:

3

các bạn có thắc mắc về đề bài hoặc test xin liên hệ quynh2538 qua forum vnoi.info

* **Người up:** huy391992
* **Điểm:** 0.31
* **Ngôn ngữ cho phép:**
* **Nguồn bài:** Anh Quỳnh bựa đòi add =))

Copying DNA ( COPYDNA )

**Giới hạn**

* **Thời gian:** 0.229s
* **Bộ nhớ:** 1536MB
* **Mã nguồn:** 50000 bytes

Cho một xâu DNA S gồm các ký tự {A, C, G, T}. Bạn sẽ làm việc trên một xâu T, ban đầu có giá trị rỗng. Tìm số thao tác sao chép nhỏ nhất để biến T thành một xâu cho trước. Biết rằng mỗi thao tác sao chép có một trong hai dạng:

* sao chép S i j k: sao chép đoạn S[i..j] vào xâu T bắt đầu từ vị trí k
* sao chép T i j k: sao chép đoạn T[i..j] vào xâu T bắt đầu từ vị trí k

Lưu ý nếu i > j có nghĩa là ta sao chép đoạn xâu theo thứ tự ngược. Mỗi ký tự trong T chỉ được tạo ra đúng một lần, nghĩa là không được sao chép đè lên ký tự đó.

Ví dụ: Với S = “ACTG” hãy tạo T = “GTACTATTATA”

1. Tạo GT......... bằng cách sao chép và đảo xâu “TG” từ S.
2. Tạo GTAC....... bằng cách sao chép “AC” từ S.
3. Tạo GTAC...TA.. bằng cách sao chép “TA” từ T.
4. Tạo GTAC...TAAT bằng cách sao chép và đảo xâu “TA” từ T.
5. Tạo GTACAATTAAT bằng cách sao chép “AAT” từ T.

Dữ liệu

Dòng đầu tiên chứa t là số bộ test. Với mỗi test có hai dòng chứa xâu S và xâu T với độ dài không quá 18.

Kết quả

Với mỗi test, in ra số thao tác sao chép ít nhất để tạo ra T từ S, hoặc in ra "impossible" nếu không thể làm được.

Ví dụ

Dữ liệu

5

ACGT

GTAC

A

C

ACGT

TGCA

ACGT

TCGATCGA

A

AAAAAAAAAAAAAAAAAA

Kết quả

2

impossible

1

4

6

* **Người up:** paulmcvn
* **Điểm:** 1.67
* **Ngôn ngữ cho phép:**
* **Nguồn bài:** NCPC 2007

Bộ ba cao thủ ( NKTRIO )

**Giới hạn**

* **Thời gian:** 0.108s
* **Bộ nhớ:** 1536MB
* **Mã nguồn:** 50000 bytes

Ở thời loạn, giang hồ có rất nhiều cao thủ võ lâm, mỗi người trong số họ lại có những tuyệt chiêu. Nếu 2 cao thủ giang hồ so tài với nhau thì từ những sở trường và sở đoản của họ, ta có thể biết trước được cao thủ nào sẽ thắng. Những cao thủ đang có ở VNOI như conankudo, gothdn, kaiel, nahnhnahk, pirate... đang muốn thi tài để xem ai được chọn làm bộ ba cao thủ.

Để mưu nghiệp lớn, minh chủ võ lâm Nuga cần tìm ra một bộ ba trong số các cao thủ giang hồ hiện tại. Để các cao thủ này quy phục dưới trướng của mình và không làm phản, Nuga muốn bộ ba cao thủ này có thể khắc chế được nhau; điều này có nghĩa là nếu 3 cao thủ được chọn là A, B và C thì A phải thắng được B, B phải thắng được C và C phải thắng được A.

Bạn hãy giúp Nuga chọn ra một bộ ba cao thủ thoả mãn yêu cầu của ông.

Dữ liệu:

Dòng đầu tiên ghi n là số cao thủ trên giang hồ (3 ≤ n ≤ 1000)

Tiếp theo là n dòng, mỗi dòng có n số. A[i,j] = 1 là người i thắng j. Dữ liệu luôn đảm bảo A[i,j] + A[j,i] = 1. A[i,i] = 0 với mọi i.

Kết quả:

Ghi ra ba số nguyên A, B và C là thứ tự của ba cao thủ thoả mãn A thắng B, B thắng C và C thắng A. Trong trường hợp có nhiều cách lựa chọn, bạn chỉ cần chỉ ra một cách; trong trường hợp không có cách lựa chọn thoả mãn yêu cầu, ghi ra ba số -1.

Ví dụ:

Dữ liệu

5

0 1 1 1 0

0 0 1 1 0

0 0 0 0 1

0 0 1 0 0

1 1 0 1 0

Kết quả

2 3 5

Dữ liệu

3

0 1 1

0 0 1

0 0 0

Kết quả

-1 -1 -1

* **Người up:** hphong
* **Điểm:** 0.16
* **Ngôn ngữ cho phép:**
* **Nguồn bài:** Test data added by Nuga

Xâu con chung dài nhất ( QBSTR )

**Giới hạn**

* **Thời gian:** 0.199s
* **Bộ nhớ:** 1536MB
* **Mã nguồn:** 50000 bytes

Xâu ký tự X được gọi là xâu con của xâu ký tự Y nếu ta có thể xoá đi một số ký tự trong xâu Y để được xâu X.

Cho biết hai xâu ký tự A và B, hãy tìm xâu ký tự C có độ dài lớn nhất và là con của cả A và B.

Input

Dòng 1: chứa xâu A

Dòng 2: chứa xâu B

Output

Chỉ gồm một dòng ghi độ dài xâu C tìm được

Example

Input:

abc1def2ghi3

abcdefghi123

Output:

10

* **Người up:** cun
* **Điểm:** 0.04
* **Ngôn ngữ cho phép:**

Lights ( LIGHTS )

**Giới hạn**

* **Thời gian:** 0.24s
* **Bộ nhớ:** 1536MB
* **Mã nguồn:** 50000 bytes

Hệ thống chiếu sáng của rạp hát có 2n bóng đèn được xếp thành 2 hàng mỗi hàng có n bóng. Mỗi bóng có 2 trạng thái là được bật sáng hoặc tắt, ban đầu tất cả các đèn đều được tắt.

Những người thợ trang trí muốn bật một số bóng đèn sao cho có thể tạo thành hình ảnh đẹp mắt. Hệ thống điều khiển chỉ cho phép ta mỗi một lần chỉ có thể đổi trạng thái từ bật sang tắt hoặc từ tắt sang bật của 2 bóng đèn ở cùng cột hoặc 1 số bóng đèn liên tiếp ở cùng hàng.

Yêu cầu:

* Cho trước trạng thái cuối cùng của hệ thống đèn mà những người thợ trang trí mong muốn.
* Tính số lần bật hoặc tắt ít nhất để được trạng thái đó.

Hình ảnh dưới đây minh họa 7 bước để đạt trạng thái cuối cùng :

**0**

00000000000000000000

00000000000000000000

**1**

**111**00000000000000000

00000000000000000000

**2**

111000 **1**0000000000000

000000 **1**0000000000000

**3**

111000 **1**0000000000000

0 **11111011**00000000000

**4**

1110110 **1111**000000000

01111101100000000000

**5**

11101101111000 **11111**0

01111101100000000000

**6**

111011011110001 **0**1110

011111011000000 **1**0000

**7**

11101101111000101 **0**10

01111101100000010 **1**00

Dữ liệu:

* Dòng đầu tiên chứa số nguyên n là số bóng đèn ở mỗi hàng ((1 ≤ n ≤ 10 000).
* 2 dòng tiếp theo, mỗi dòng là trạng thái của mỗi hàng đèn tại thời điểmcuối cùng, số 1 biểu thị 1 bóng đèn đang sáng và 0 là 1 bóng đèn đang tắt.

Kết quả:

* Đưa ra số lần bật hoặc tắt ít nhất để được trạng thái theo yêu cầu.

Ví dụ:

Dữ liệu :

20

11101101111000101010

01111101100000010100

Kết quả :

7

* **Người up:** aukcwe
* **Điểm:** 0.48
* **Ngôn ngữ cho phép:**
* **Nguồn bài:** Croatian Regional Competition in Informatics 2006

Đếm chuỗi đối xứng ( QBPAL )

**Giới hạn**

* **Thời gian:** 0.163s
* **Bộ nhớ:** 1536MB
* **Mã nguồn:** 50000 bytes

Trong một buổi học viết chữ, Bờm phát hiện trong một số từ khi bỏ đi một số ký tự thì đọc ngược hay đọc xuôi đều giống nhau.

Ví dụ từ IOICAMP, khi xóa đi các chữ cái C,A,M,P, thì còn lại IOI là một từ đối xứng.

Bờm cảm thấy thú vị, và cậu tiếp tục thử xóa các ký tự khác, kết quả là có thêm nhiều từ đối xứng nữa: II, I, O, C… Nhưng nếu với một từ dài, cứ thử từng cách xóa như vậy thì thật mất thời gian. Bạn hãy viết chương trình giúp Bờm tính số cách xóa sao cho từ thu được đối xứng. Hai cách xóa chỉ khác nhau bởi thứ tự xóa các ký tự thì coi như trùng nhau.

Input

Một dòng duy nhất là từ cần tính số cách xóa, từ này chỉ chứa các chữ cái in hoa A, B, .., Z. ( Độ dài từ không quá 120 )

Output

Một số duy nhất là số cách xóa.

Example

Input:

IOICAMP

Output:

9

* **Người up:** cun
* **Điểm:** 0.14
* **Ngôn ngữ cho phép:**
* **Nguồn bài:** Hứa Lê Thanh Vy

Đến trường ( QBSCHOOL )

**Giới hạn**

* **Thời gian:** 0.148s
* **Bộ nhớ:** 1536MB
* **Mã nguồn:** 50000 bytes

Ngày 27/11 tới là ngày tổ chức thi học kỳ I ở trường ĐH BK. Là sinh viên năm thứ nhất, Hiếu không muốn vì đi muộn mà gặp trục trặc ở phòng thi nên đã chuẩn bị khá kỹ càng. Chỉ còn lại một công việc khá gay go là Hiếu không biết đi đường nào tới trường là nhanh nhất.

Thường ngày Hiếu không quan tâm tới vấn đề này lắm cho nên bây giờ Hiếu không biết phải làm sao cả . Bản đồ thành phố là gồm có N nút giao thông và M con đường nối các nút giao thông này. Có 2 loại con đường là đường 1 chiều và đường 2 chiều. Độ dài của mỗi con đường là một số nguyên dương.

Nhà Hiếu ở nút giao thông 1 còn trường ĐH BK ở nút giao thông N. Vì một lộ trình đường đi từ nhà Hiếu tới trường có thể gặp nhiều yếu tố khác như là gặp nhiều đèn đỏ , đi qua công trường xây dựng, ... phải giảm tốc độ cho nên Hiếu muốn biết là có tất cả bao nhiêu lộ trình ngắn nhất đi từ nhà tới trường. Bạn hãy lập trình giúp Hiếu giải quyết bài toán khó này.

Input

Dòng thứ nhất ghi hai số nguyên N và M.

M dòng tiếp theo, mỗi dòng ghi 4 số nguyên dương K, U, V, L. Trong đó:

K = 1 có nghĩa là có đường đi một chiều từ U đến V với độ dài L.

K = 2 có nghìa là có đường đi hai chiều giữa U và V với độ dài L.

Output

Ghi hai số là độ dài đường đi ngắn nhấn và số lượng đường đi ngắn nhất. Biết rằng số lượng đường đi ngắn nhất không vượt quá phạm vì int64 trong pascal hay long long trong C++.

Example

Input:

3 2

1 1 2 3

2 2 3 1

Output:

4 1

Giới hạn:

1 ≤ N ≤ 5000

1 ≤ M ≤ 20000

Độ dài các con đường ≤ 32000

* **Người up:** cun
* **Điểm:** 0.08
* **Ngôn ngữ cho phép:**
* **Nguồn bài:** Nguyễn Minh Hiếu

Đếm cây nhị phân ( QBTREEK )

**Giới hạn**

* **Thời gian:** 0.342s
* **Bộ nhớ:** 1536MB
* **Mã nguồn:** 50000 bytes

Hãy đếm số cây nhị phân khác nhau thỏa mãn điều kiện:

Số nút của cây là N.

Mỗi nút có số nút con là 0 hoặc 2.

Chiều cao của cây bằng K.

Hai cây nhị phân được coi là khác nhau nếu cấu trúc của chúng là khác nhau. Biết rằng:

Hai cây gồm 0 nút được coi là giống nhau.

Hai cây X, Y giống nhau nếu cây con trái của X giống cây con trái của Y và cây con phải của X giống cây con phải của Y.

Input

Một dòng duy nhất ghi số N và K. ( 1 <= n <= 1000, 1 <= k <= 500 ).

Output

Số cây nhị phân khác nhau sau khi MOD 9901.

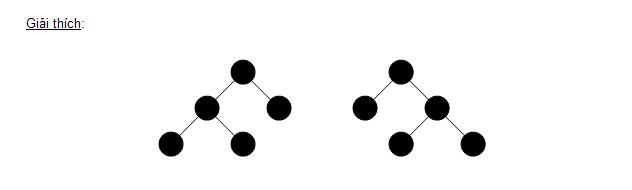
Example

Input:

5 3

Output:

2



* **Người up:** cun
* **Điểm:** 0.73
* **Ngôn ngữ cho phép:**
* **Nguồn bài:** ACM

Các đại lý ( QBAGENTS )

**Giới hạn**

* **Thời gian:** 0.039s
* **Bộ nhớ:** 1536MB
* **Mã nguồn:** 50000 bytes

Sau một số rủi ro và thất bại trong kinh doanh, tổng giám đốc công ty Fsoft là Zone quyết định tổ chức cho các sếp nhỏ của các đại lý thuộc công ty gặp mặt và thảo luận với nhau. Công ty Fsoft là một công ty cực kì lớn trải khắp toàn cầu nên một vấn đề lớn đặt ra là làm sao tổ chức cho 2 sếp nhỏ gặp nhau trong thời gian sớm nhất. Vấn đề đặc biệt trở nên hóc búa vì các nhân viên của công ty chỉ được đi bằng mạng giao thông của công ty để đảm bảo an toàn, bảo mật và chi phí. Nhưng mạng này lại hơi tệ:

- Các nhân viên buộc phải di chuyển theo các tuyến giao thông giữa các đại lý.

- Mạng giao thông của công ty là mạng gồm các tuyến đường 1 chiều.

- Các nhân viên khi đi trong mạng thì mỗi giờ đi được theo đúng 1 tuyến đường và phải liên tục di chuyển (nghĩa là không được dừng lại).

Được cái đây là mạng nội bộ và với công nghệ đỉnh cao nên không có chuyện tắc đường. Vì vậy, trong 1 giờ luôn có thể di chuyển từ đại lý này sang đại lý khác nếu có đường.

Zone muốn nhân viên của mình không lãng phí thời gian. Bởi vậy ông muốn tính thời gian ngắn nhất mà 2 sếp ở 2 đại lý cho trước có thể gặp nhau. Đáng tiếc là Zone chỉ giỏi kinh doanh, còn lập trình thì quá yếu kém. Bạn là nhân viên dưới quyền Zone và đang rất muốn thể hiện khả năng của mình. Vậy thì, hãy nhân cơ hội này để cho Zone thấy trình độ tuyệt vời của bạn.

Input

Dòng đầu ghi 2 số N, M là số đại lý và số tuyến đường trong mạng giao thông của công ty Fsoft. (N ≤ 250)

Dòng thứ 2 ghi S,T lần lượt là số thứ tự 2 đại lý có 2 sếp cần phải gặp nhau.

M dòng tiếp theo mỗi dòng ghi 2 số nguyên U, V thể hiện có đường đi một chiều từ U tới V.

Output

Gồm một dòng duy nhất ghi thời gian nhỏ nhất 2 sếp có thể gặp nhau.

Nếu 2 sếp không thể gặp nhau ghi -1.

Example

Input:

6 7

1 5

1 2

4 5

2 3

3 4

4 1

5 4

5 6

Output:

3

* **Người up:** cun
* **Điểm:** 0.29
* **Ngôn ngữ cho phép:**

Bộ ba điểm thẳng hàng ( QBPOINT )

**Giới hạn**

* **Thời gian:** 0.354s
* **Bộ nhớ:** 1536MB
* **Mã nguồn:** 50000 bytes

Trong các cuộc thi tin học, sự xuất hiện của những bài toán hình học làm đội tuyển CBQ khá lúng túng. Do đó thầy Thạch quyết định cho đội tuyển luyện tập các bài toán hình học. Bắt đầu từ điểm, thầy đưa ra bài toán:

Cho n điểm trong mặt phẳng Oxy, hãy đếm số bộ 3 điểm thằng hàng

Input

Dòng thứ nhất ghi số N là số điểm trên mặt phẳng.

N dòng tiếp theo, mỗi dòng ghi tọa độ của một điểm.

Output

Một số duy nhất là số bộ 3 điểm thẳng hàng.

Example

Input:

6

0 0

0 1

0 2

1 1

2 0

2 2

Output:

3

Giới hạn:

1 ≤ N ≤ 2000.

Tọa độ các điểm có trị tuyệt đối không quá 10000.

* **Người up:** cun
* **Điểm:** 0.22
* **Ngôn ngữ cho phép:**
* **Nguồn bài:** Nguyễn Hoành Tiến

Chip &amp; Dale ( QBWIRE )

**Giới hạn**

* **Thời gian:** 0.029s
* **Bộ nhớ:** 1536MB
* **Mã nguồn:** 50000 bytes

Hai anh em Chip và Dale đang chơi trò chơi “nối dây”. Đó là trò chơi trên bảng có N × N nút lưới. Các nút của lưới được đánh số từ 0 đến N – 1 theo chiều từ trên xuống và 0 đến N – 1 theo chiều từ trái sang phải. Tọa độ của một nút được thể hiện bằng một cặp số trong đó số thứ nhất là tọa độ cột, số thứ hai là tọa độ dòng. Chip sẽ chọn ra 4 điểm A 1, A 2, B 1, B 2. Chip yêu cầu Dale sử dụng các nét vẽ ngang (dọc) theo các đường kẻ của lưới để vẽ 2 đường gấp khúc thỏa mãn:

- Đường gấp khúc thứ nhất nối A 1với A 2.

- Đường gấp khúc thứ hai nối B 1với B 2.

- Hai đường gấp khúc không có điểm chung.

- Tổng độ dài hai đường gấp khúc là nhỏ nhất có thể.

Dale có vẻ không thành thạo lắm với trò chơi này, bạn hãy giúp Dale tính tổng độ dài nhỏ nhất của hai đường gấp khúc.

Input

Dòng đầu ghi số nguyên T là số test.

Tiếp theo là T test, mỗi test được ghi trên 3 dòng

Dòng thứ nhất ghi số N.

Dòng thứ hai ghi 4 số là tọa độ cột của 4 điểm A 1, A 2, B 1, B 2.

Dòng thứ ba ghi 4 số là tọa độ dòng của 4 điểm A 1, A 2, B 1, B 2.

Output

Mỗi test ghi đáp số trên một dòng.

Nếu không có cách nối dây thỏa mãn, ghi ra -1.

Nếu có cách nối dây thỏa mãn, ghi ra tổng độ dài 2 đường gấp khúc.

Example

Input:

1

7

2 5 4 4

1 4 5 0

Output:

15

Giới hạn:

1 ≤ T ≤ 5

2 ≤ N ≤ 500

* **Người up:** cun
* **Điểm:** 0.87
* **Ngôn ngữ cho phép:**

Tặng hoa ( QBFLOWER )

**Giới hạn**

* **Thời gian:** 0.206s
* **Bộ nhớ:** 1536MB
* **Mã nguồn:** 50000 bytes

Sau kì thi Marathon, thầy My đã quyết định tổ chức một buổi dạ hội nho nhỏ cho các thí sinh. Trong buổi dạ hội này sẽ có N bạn nữ và M bạn nam. Để không khí thêm phần vui vẻ thì thầy My đã nghĩ ra tiết mục các bạn nam tặng hoa cho các bạn nữ. Mỗi bạn nam sẽ đưa cho ban tổ chức danh sách 2 bạn nữ mà bạn đó muốn tặng hoa nhất. Tuy nhiên tiền tài trợ cho buổi dạ hội không còn nhiều (vì đã phải dành hầu hết để trao giải thưởng). Nhưng ban tổ chức cũng không muốn bạn nữ nào không được nhận hoa. Thầy My đã giao việc này cho Mr.Hải Minh, và anh ta đang rất bối rối vì không biết làm thế nào.

Bạn hãy giúp Mr. Hải Minh chọn ra ít bạn nam nhất đứng ra đại diện cho các bạn nam để tặng hoa các bạn nữ sao cho bạn nữ nào cũng được tặng hoa. Biết rằng mỗi bạn nam được chọn sẽ tặng hoa cho cả hai bạn nữ trong danh sách của bạn đó.

Input

Dòng đầu ghi hai số N và M. (2 ≤ N ≤ 1000, 1 ≤ M ≤ 1000)

Dòng thứ i trong M dòng tiếp theo ghi hai số ai và bi là hai bạn nữ mà bạn nam thứ i muốn tặng hoa.

Output

Số bạn nam ít nhất cần lựa chọn

Example

Input:

3 3

1 2

2 3

1 3

Output:

2

* **Người up:** cun
* **Điểm:** 1.23
* **Ngôn ngữ cho phép:**
* **Nguồn bài:** IOICAMP

Trò chơi ở hội chợ ( QBMOUSE )

**Giới hạn**

* **Thời gian:** 0.05s
* **Bộ nhớ:** 1536MB
* **Mã nguồn:** 50000 bytes

Ở một phiên chợ dành cho thiếu nhi có một trò chơi rất hấp dẫn với những con chuột trong một cái bàn. Nhiệm vụ của người chơi là phải làm bị thương những con chuột, càng nhiều càng tốt với một chiếc búa. Để làm cho trò chơi dễ dàng hơn, các bạn nhỏ đã hỏi bụt để biết được vị trí cũng như thời gian chính xác mà mỗi con chuột xuất hiện.

Coi mặt phẳng của mặt bàn là một hệ tọa độ Đề-các, mỗi con chuột sẽ xuất hiện từ một lỗ thủng trên mặt bàn với tọa độ nguyên (x,y) thỏa mãn 0 ≤ x, y ≤ N. Trong mỗi thời điểm, một vài con chuột sẽ xuất hiện ở những lỗ thủng khác nhau và sau đó lại chui xuống vào trước thời điểm tiếp theo. Ngay sau khi các con chuột xuất hiện và ngay trước lúc chúng biến mất, người chơi có thể di chuyển chiếc búa trên một đoạn thẳng có chiều dài tối đa là D. Để đơn giản, tọa độ của chiếc búa trước và sau khi di chuyển luôn là nguyên và coi kích thước của chiếc búa không đáng kể. Một con chuột sẽ bị thương nếu tâm của lỗ mà con chuột xuất hiện nằm trên đoạn thẳng chiếc búa di chuyển. Khi bắt đầu trò chơi, ngay trước thời điểm đầu tiên mà những con chuột xuất hiện, người chơi có thể đặt búa ở bất cứ vị trí nào.

Hãy giúp các em thiếu nhi làm bị thương nhiều chuột nhất có thể, để các em được những phần quà của ban tổ chức. Và tất nhiên ai giúp đỡ các em thiếu nhi nhiều nhất cũng sẽ có phần thưởng là điểm số trong tuần thi này.

Input

Dòng đầu ghi 3 số N, D, M trong đó M là số con chuột.

M dòng tiếp theo mỗi dòng ghi 3 số lần lượt là tọa độ và thời điểm mà mỗi con chuột xuất hiện.

Output

Xuất ra một số duy nhất là số chuột lớn nhất có thể làm bị thương.

Example

Input:

4 5 4

0 0 1

1 0 1

0 1 2

1 2 2

Output:

4

Giải thích:

Đầu tiên búa ở 1 0.

Bước 1 đập 2 con chuột ở 1 0, 0 0, rồi đưa búa tiếp đến -1 0.

Bước 2 đập 2 con chuột ở 0 1, 1 2.

Giới hạn:

1 ≤ N ≤ 20

1 ≤ D ≤ 5

1 ≤ M ≤ 1000

Thời điểm mà mỗi con chuột xuất hiện là số tự nhiên không lớn hơn 10.

* **Người up:** cun
* **Điểm:** 1.10
* **Ngôn ngữ cho phép:**

Cổ phiếu ( QBSTOCK )

**Giới hạn**

* **Thời gian:** 0.136s
* **Bộ nhớ:** 1536MB
* **Mã nguồn:** 50000 bytes

Qua nhiều năm hoạt động, IOICamp đã tập hợp được nhiều tài năng tin học và qua đó dần dần trở thành một tập đoàn công nghệ lớn. Đến năm 2222, IOICamp đã chính thức trở thành một tập đoàn lớn có cổ phiếu trên thị trường chứng khoán. Tuy nhiên để tồn tại được trên thị trường chứng khoán thì việc nắm bắt được tình hình lên xuống của giá cổ phiếu là vô cùng cần thiết. Để làm được việc này, IOICamp rất cần đến các chuyên gia tin học của mình. Sau nhiều năm nghiên cứu các thành viên của IOICamp đã thành công trong việc xấp xỉ sự biến động của cổ phiếu thành một hàm đa thức với biến thời gian. Tuy nhiên công việc khó khăn trước mắt là phải tìm được thời điểm cổ phiếu có giá thấp nhất và cao nhất và IOICamp muốn nhờ các bạn làm giúp việc này.

Các bạn sẽ được cung cấp hàm đa thức mô tả biến động của giá cổ phiếu trong một khoảng thời gian cho trước. Hàm đa thức này có dạng:

f(t) = a.t 4+ b.t 3+ c.t 2+ d.t + e

Với a, b, c, d, e là các hệ số nguyên cho trước. Hãy tìm giá cổ phiếu cao nhất f maxvà giá cổ phiếu thấp nhất f mintrong khoảng thời gian từ t 1đến t 2(nghĩa là t 1≤ t ≤ t 2).

**Đ ây là 1 bài khá đơn giản cho các bạn đã qua môn Phương Pháp Tính :D**

Input

Dòng thứ nhất ghi 5 số nguyên a, b, c, d, e.

Dòng thứ hai ghi 2 số nguyên t 1, t 2

Output

Dòng đầu tiên ghi số thực f maxlà giá cổ phiếu cao nhất.

Dòng thứ hai ghi số thực f minlà giá cổ phiếu thấp nhất.

Example

**Input:**

1 0 -2 0 1

0 2

**Output:**

9.000000000

0.000000000

**Giới hạn:**

|a|, |b|, |c|, |d|, |e| ≤ 10

–20 ≤ t1 < t2 ≤ 20

Kết quả ghi ra với độ chính xác 9 chữ số sau dấu phẩy

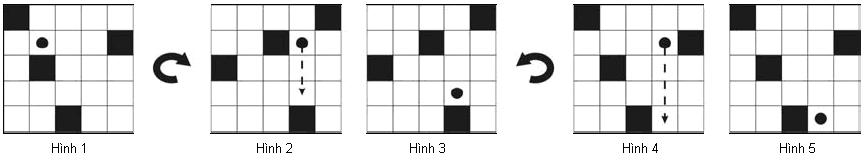
* **Người up:** cun
* **Điểm:** 1.08
* **Ngôn ngữ cho phép:**
* **Nguồn bài:** IOICAMP4

Trò chơi trí tuệ ( IQ )

**Giới hạn**

* **Thời gian:** 0.1s
* **Bộ nhớ:** 1536MB
* **Mã nguồn:** 50000 bytes

Hình thức kiểm tra IQ hiện nay mà người ta vẫn hay dùng đang cho thấy những khuyết điểm lớn. Đó là sự dập khuôn, không đa dạng. Một người chỉ sau vài lần làm thử các bài test IQ sẽ quen với các dạng đề bài như dãy quy luật, chọn hình sai v.v... Như vậy thì bài thi IQ sẽ không còn ý nghĩa kiểm tra độ thông minh, sáng tạo của con người nữa. Để khắc phục người ta vừa phát minh ra một hình thức kiểm tra chỉ số IQ mới. Đó là một trò chơi với một bảng gồm M dòng và N cột. Để thuận tiện, người ta đánh số các dòng từ 1 đến M theo chiều từ trên xuống, các cột từ 1 đến N theo chiều từ trái sang phải. Một ô của bảng sẽ được thể hiện bằng một cặp số nguyên (u, v) trong đó u là chỉ số dòng, v là chỉ số cột. Trong số M × N ô của bảng, sẽ có một số ô có chướng ngại vật. Ban đầu, tại một số ô, người ta để sẵn một số viên bi. Nhiệm vụ của bạn là đưa các viên bi đó về một số ô đích. Để làm được điều đó, bạn được sử dụng các phép quay bảng sang trái/phải (ngược chiều kim đồng hồ/thuận chiều kim đồng hồ). Các phép quay có thể được mô tả như sau:



Giải thích:

Hình 1 sau khi quay phải (thuận chiều kim đồng hồ) được hình 2.

Ở hình 2, do trọng lực, viên bi rơi xuống. Ta có bảng như hình 3.

Hình 3 sau khi quay trái (ngược chiều kim đồng hồ) được hình 4.

Ở hình 4, do trọng lực, viên bi rơi xuống. Ta có bảng như hình 5.

Yêu cầu:

Đề thi IQ không bao giờ đơn giản. Trong bảng M × N, sẽ không chỉ có một viên bi mà sẽ có K viên bi. Bạn sẽ phải đưa K viên bi này đến K ô đích bằng ít lần xoay bảng nhất. Cũng nói thêm rằng, trong quá trình xoay bảng và quá trình các viên bi rơi xuống, không có hai viên bi nào đồng thời ở một ô. Nếu hai viên bi cùng trên một cột rơi xuống, hai viên bi sẽ nằm ở hai ô liên tiếp, trên dưới nhau.

Input

Dòng đầu ghi ba số M, N và K.

K dòng tiếp theo, mỗi dòng ghi một cặp số là tọa độ của một ô đích.

M dòng tiếp theo, mỗi dòng ghi một xâu độ dài N thể hiện trạng thái ban đầu của bảng. Xâu gồm các kí tự “#” thể hiện ô chướng ngại vật, “.” thể hiện ô trống, “o” thể hiện một viên bi.

Dữ liệu vào đảm bảo tại thời điển ban đầu, các viên bi luôn được đặt phía trên một ô chướng ngại vật hoặc ở hàng dưới cùng của bảng. Dữ liệu vào cũng đảm bảo luôn có ít nhất một cách làm.

Output

Gồm 1 dòng duy nhất ghi P là số lần xoay bảng ít nhất

Example

**Input:**

5 5 1

5 4

#....

.o..#

.#...

.....

..#..

**Output:**

2

**Giới hạn:**

1 ≤ M, N ≤ 10

1 ≤ K ≤ 4

* **Người up:** cun
* **Điểm:** 0.99
* **Ngôn ngữ cho phép:**
* **Nguồn bài:** Lê Ðôn Khuê

Xây nhà ( HOUSE )

**Giới hạn**

* **Thời gian:** 0.117s
* **Bộ nhớ:** 1536MB
* **Mã nguồn:** 50000 bytes

Cùng với phần thưởng là tấm ảnh kèm chữ kí tặng của hoa hậu Việt Nam 2006 – Mai Phương Thúy, BB còn được nhận được một khoản tiền. BB quyết định sẽ dùng khoản tiền này để xây ngôi nhà mơ ước của mình. Hiện nay, BB đã có một mảnh đất được chia thành M x N ô. BB muốn xây ngôi nhà của mình trên mảnh đất này. BB muốn rằng ngôi nhà của mình phải là ngôi nhà thật đặc biệt. Nó sẽ phải đáp ứng đủ các yêu cầu:

Là tập hợp của một số ô liên thông. Hai ô được coi là kề nhau nếu chúng có chung cạnh.

Có ít nhất hai trục đối xứng: một trục ngang, một trục dọc (cả hai trục đều phải song song với cạnh của khu đất). BB vốn tin vào thuyết phong thủy mà.

BB lại là người cẩn thận, nên cậu muốn rằng ngôi nhà của mình sẽ có ít nhất một điểm có thể đặt camera để quan sát toàn bộ ngôi nhà. Một điểm A được coi là có thể quan sát được điểm B nếu đoạn thẳng AB nằm hoàn toàn bên trong ngôi nhà hoặc có điểm chung với tường ngôi nhà, nhưng không được đi ra bên ngoài ngôi nhà.

Với mảnh đất của mình, bạn hãy giúp BB tính xem có bao nhiêu cách để xây được ngôi nhà mơ ước.

Input

Dòng thứ nhất ghi hai số nguyên dương M, N.

Output

Gồm một số duy nhất là phần dư của phép chia số cách xây dựng ngôi nhà cho 12051987.

Example

Input:

2 2

Output:

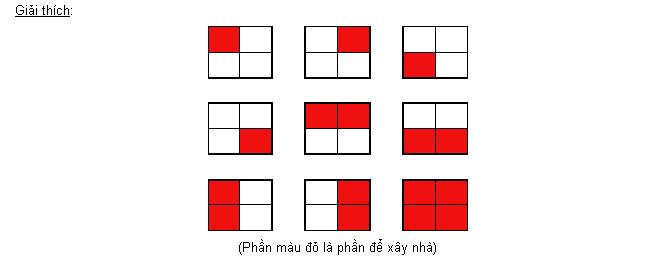
9

Giới hạn:

M, N ≤ 1000

Thời gian: 1 s/test

Có 30% số test có M, N ≤ 5



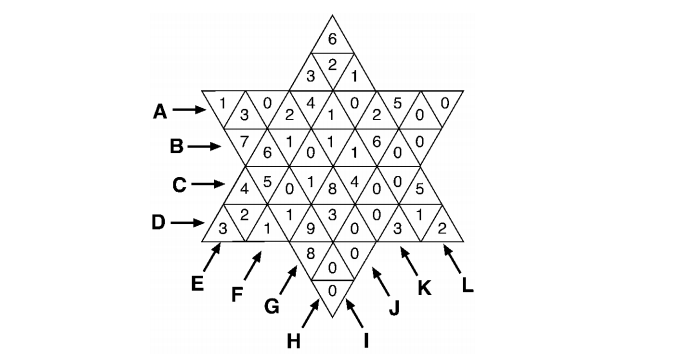
* **Người up:** cun
* **Điểm:** 0.82
* **Ngôn ngữ cho phép:**
* **Nguồn bài:** IOICAMP4

Bảng số hình sao ( QBSTAR )

**Giới hạn**

* **Thời gian:** 1.639s
* **Bộ nhớ:** 1536MB
* **Mã nguồn:** 50000 bytes

BB có một bảng số gồm 48 ô. Mỗi ô được ghi một chữ số từ 0 đến 9. Mỗi ô thuộc 2 hàng hoặc 3 hàng như hình vẽ bên. Có 12 hàng được đánh số từ A đến L. Ví dụ ô ghi số 9 thuộc 3 hàng D, G và I.



Sau khi điền toàn bộ số vào bảng, BB ghi lại 12 số lớn nhất của 12 hàng. Ví dụ hàng A có số lớn nhất là 5, hàng B có số lớn nhất là 7, hàng H có số lớn nhất là 0. Sau đó BB xóa toàn bộ bảng đi. BB sẽ vẽ lại hai bảng số khác đảm bảo 12 số lớn nhất của 12 hàng giống bảng ban đầu. BB muốn rằng hai bảng số mới vẽ: bảng thứ nhất sẽ có tổng các số trong 48 ô là nhỏ nhất, bảng thứ hai sẽ có tổng các số trong 48 ô là lớn nhất. Bạn hãy giúp BB thực hiện điều này.

Input

Một dòng duy nhất gồm 12 số lần lượt là số lớn nhất của 12 hàng trong bảng ban đầu.

Output

Gồm hai số, số thứ nhất là tổng các số trong 48 ô nhỏ nhất có thể được, số thứ hai là tổng các số lớn nhất có thể được.

Nếu không có cách nào điền được, in ra hai số -1.

Example

**Input:**

5 7 8 9 6 1 9 0 9 8 4 6

**Output:**

40 172

* **Người up:** cun
* **Điểm:** 1.60
* **Ngôn ngữ cho phép:**

Cắt bánh pizza ( QBPIZZA )

**Giới hạn**

* **Thời gian:** 0.075s
* **Bộ nhớ:** 1536MB
* **Mã nguồn:** 50000 bytes

Để chuẩn bị cho buổi giao lưu của cuộc thi HAOI năm nay, msn và Lola đã mua về một chiếc bánh pizza có hình thù đặc biệt. Chiếc bánh là một hình đa giác N đỉnh không tự cắt.

BB được giao nhiệm vụ sẽ cắt chiếc bánh này để chia cho mọi người. Để tăng thêm sự thú vị cho buổi giao lưu, BB đưa ra bài toán sau cho các bạn thí sinh. Nếu ta đặt chiếc bánh vào hệ tọa độ Đề các, mỗi nhát cắt sẽ là một đường thẳng song song với trục hoành.

BB muốn biết với mỗi nhát cắt như vậy, tổng độ dài của nhát cắt đi qua chiếc bánh là bao nhiêu. Một nhát cắt nếu đi trùng vào một cạnh của chiếc bánh phải tính cả độ dài của cạnh đó (xem trường hợp nhát cắt y = 3 trong test ví dụ). Người trả lời đúng nhiều nhất nhận được phần bánh to nhất, nhiều pho mát nhất, nhiều thịt bò nhất. beo\_chay\_so vốn rất ham ăn và rất muốn có phần bánh ngon nhất đó. Bạn hãy giúp beo\_chay\_so nhé.

Input

Dòng đầu tiên là hai số nguyên dương N và M trong đó M là số nhát cắt sẽ thực hiện.

Dòng thứ i trong N dòng tiếp theo ghi hai số nguyên X i, Y ilà tọa độ của đỉnh thứ i của chiếc bánh.

Tọa độ đỉnh của chiếc bánh sẽ được liệt kê theo chiều kim đồng hồ.

Trên M dòng cuối, mỗi dòng có một số nguyên alpha trong đó đường thẳng y = alpha là đường thẳng thể hiện nhát cắt.

Output

Tương ứng với M dòng trong input, hãy ghi ra M số trên M dòng thể hiện tổng độ dài của nhát cắt thứ i đi qua chiếc bánh. Các số được ghi với độ chính xác ít nhất 6 chữ số sau dấu phẩy.

Example

Input:

8 4

0 0

0 3

1 3

3 1

3 3

4 1

3 0

1 1

0

1

2

3

Output:

0.000000

4.000000

2.500000

1.000000

Giới hạn:

1 ≤ N ≤ 1000

1 ≤ M ≤ 50000

Các tọa độ có trị tuyệt đối không quá 105

* **Người up:** cun
* **Điểm:** 1.57
* **Ngôn ngữ cho phép:**
* **Nguồn bài:** HAOI

Cắt bánh ( QBCAKE )

**Giới hạn**

* **Thời gian:** 0.143s
* **Bộ nhớ:** 1536MB
* **Mã nguồn:** 50000 bytes

Sau cuộc thi HAOI lần này, Ban tổ chức sẽ làm một chiếc bánh khổng lồ có hình một đa giác lồi để chiêu đãi các thí sinh. Trưởng Ban tổ chức quyết định luật cắt bánh như sau: Mỗi lần chọn 1 đỉnh của đa giác rồi cắt bỏ đỉnh đó bằng cách cắt qua 2 đỉnh kề bên. Phần bánh hình tam giác có được từ mỗi lần cắt như vậy sẽ chia cho các thí sinh. Công việc cắt bánh sẽ tiếp tục cho đến khi chiếc bánh có dạng một tứ giác. Miếng bánh cuối cùng này sẽ giành cho người cắt bánh.

Anh beo\_chay\_so là một thành viên của Ban tổ chức nên anh đã biết trước kế hoạch này. Vì vậy anh ta đang tính kế để chiếm được một phần bánh lớn nhất bằng cách tình nguyện làm người cắt bánh.

Yêu cầu: Hãy tính xem phần diện tích lớn nhất của miến bánh hình tứ giác mà anh béo có thể có được là bao nhiêu.

Input

Dòng thứ nhất ghi số N là số đỉnh của đa giác.

N dòng tiếp theo là các cặp số nguyên biểu diễn các đỉnh của đa giác. Các đỉnh được liệt kê cùng chiều hoặc ngược chiều kim đồng hồ

Output

Gồm 1 số duy nhất ghi diện tích lớn nhất của tứ giác tìm được. Kết quả lấy chính xác tới 1 chữ số phần thập phân.

Example

**Input:**

6

2 1

2 3

5 7

8 3

8 1

5 0

**Output:**

21.0

**Giới hạn:**

4 ≤ N ≤ 1500

|xi|, |yi| ≤ 15000

* **Người up:** cun
* **Điểm:** 0.75
* **Ngôn ngữ cho phép:**
* **Nguồn bài:** HAOI

Tổng vector ( VECTOR )

**Giới hạn**

* **Thời gian:** 0.272s
* **Bộ nhớ:** 1536MB
* **Mã nguồn:** 50000 bytes

Trong mặt phẳng tọa độ có N véc tơ. Mỗi một véc tơ được cho bởi hai chỉ số x và y. Tổng của hai véc tơ (xi, y i) và (x j, y j) được định nghĩa là một véc tơ (x i+ x j, y i+ y j). Bài toán đặt ra là cần chọn một số véc tơ trong N véc tơ đã cho sao cho tổng của các vec tơ đó là véc tơ (U, V).

Yêu cầu: Đếm số cách chọn thoả mãn yêu cầu bài toán đặt ra ở trên.

Input

Dòng thứ nhất ghi số N (0 ≤ N ≤ 30).

N dòng tiếp theo, dòng thứ i ghi các số nguyên xi, yi lần lượt là hai chỉ số của véc tơ thứ i. (|x i|, |y i| ≤ 100).

Dòng cuối cùng ghi số hai số nguyên U V (|U|, |V| ≤ 10 9).

Output

Gồm một số duy nhất là số cách chọn thoả mãn.

Example

Input:

4

0 0

-1 2

2 5

3 3

2 5

Output:

4

* **Người up:** cun
* **Điểm:** 0.11
* **Ngôn ngữ cho phép:**
* **Nguồn bài:** HAOI

Diện tích các hình tròn ( SCIRCLE )

**Giới hạn**

* **Thời gian:** 0.612s
* **Bộ nhớ:** 1536MB
* **Mã nguồn:** 50000 bytes

Sau khi đội tuyển CQB đã nắm khá chắc phần điểm và làm tốt các bài toán về điểm trong hình học. Thầy Thạch chuyển sang giải quyết các bài toán về hình tròn, dạng này thường rất có trong những cuộc thi lớn. Thầy ra cho các bạn 1 bài như sau:

Trên 1 tờ giấy thầy vẽ n đường tròn, sau đó thầy tô màu n hình tròn này.

Yêu cầu: Hãy tính diện tích của phần được tô màu.

Input

Dòng thứ nhất ghi số nguyên dương N là số hình tròn.

Dòng thứ i trong N dòng tiếp theo ghi 3 số nguyên X i, Y ivà R ilà tọa độ và bán kính của đường tròn thứ i trong N đường tròn.

Output

Ghi ra duy nhất 1 số thực là diện tích phần giấy phải tô màu. Chính xác tới 5 chữ số sau dấu phẩy ( không làm tròn, cắt tại chữ số thứ 5)

Example

Input:

2

5 6 3

5 5 5

Output:

78.53981

Giới hạn:

1 ≤ N ≤ 50

Kích thước tờ giấy là 10000 \* 10000

Tất cả các hình tròn nằm trong tờ giấy.

* **Người up:** cun
* **Điểm:** 1.63
* **Ngôn ngữ cho phép:**
* **Nguồn bài:** Hoành Tiến

Đánh số đồ thị ( GRNUM )

**Giới hạn**

* **Thời gian:** 2.0s
* **Bộ nhớ:** 1536MB
* **Mã nguồn:** 50000 bytes

Cho một đơn đồ thị vô hướng gồm K x N đỉnh, các đỉnh được chia thành K nhóm, mỗi nhóm có N đỉnh. Các nhóm được đặt tên bằng các chữ cái in hoa A, B, C, ... các đỉnh tương ứng thuộc các nhóm được đặt tên bằng các số từ 0 đến N – 1. Giả sử Ch Klà chữ cái ứng với nhóm thứ K, ta quy ước chữ cái tiếp sau A là B, tiếp sau B là C, … tiếp sau Ch Klà A và ký hiệu chữ cái tiếp sau Ch là next(Ch). Đồ thị này có các tính chất sau:

Giữa các đỉnh thuộc cùng một nhóm không có cạnh nối.

Các đỉnh thuộc 2 nhóm bất kỳ có tên là Ch và next(Ch) có đúng N cạnh nối từ đỉnh thuộc nhóm Ch đến đỉnh thuộc nhóm next(Ch).

Bạn cần được yêu cầu đánh số các đỉnh của đồ thị sao cho:

Các đỉnh thuộc 1 nhóm được đánh các số là hoán vị của các số tự nhiên từ 0 đến N – 1.

Với 2 nhóm Ch và next(Ch) bất kỳ thì N số trên N cạnh nối các đỉnh thuộc chúng là khác nhau. Nếu đỉnh i thuộc nhóm Ch được đánh số là P kề với đỉnh j thuộc nhóm next(Ch) được đánh số là Q thì cạnh nối 2 đỉnh này được đánh số là (N + P – Q) mod N.

Biết rằng 2 cách đánh số các đỉnh của đồ thị được coi là khác nhau nếu trong 2 cách đánh số, tồn tại một đỉnh thuộc một nhóm nào đó được đánh số khác nhau. Bạn hãy tính số cách đánh số khác nhau.

Input

Dòng thứ nhất ghi 2 số nguyên dương K và N là số nhóm và số đỉnh thuộc 1 nhóm.

Mỗi dòng trong K x N dòng tiếp theo ghi 1 cạnh của đồ thị theo dạng Ch i j trong đó Ch là một ký tự, i và j là 2 số nguyên với ý nghĩ có cạnh nối từ đỉnh i của nhóm Ch đến đỉnh j của nhóm next(Ch).

Output

Ghi ra duy nhất một số nguyên là số lượng cách đánh số khác nhau tìm được.

Example

Input:

3 3

A 0 0

A 1 2

A 2 1

B 1 0

B 1 2

B 2 2

C 0 2

C 1 1

C 1 2

Output:

54

Giới hạn:

1 ≤ K ≤ 5

1 ≤ N ≤ 20

Số cách đánh số khác nhau luôn đảm bảo không quá 1000000

* **Người up:** cun
* **Điểm:** 2.00
* **Ngôn ngữ cho phép:**
* **Nguồn bài:** Ioicamp - marathon 06 - 07

Phòng cháy ( FIRE )

**Giới hạn**

* **Thời gian:** 0.208s
* **Bộ nhớ:** 1536MB
* **Mã nguồn:** 50000 bytes

Để đối phó với tình hình biến động của giá xăng dầu, nước X quyết định xây dựng một kho dự trữ dầu với quy mô cực lớn. Kho chứa dầu sẽ bao gồm N bể chứa dầu hình trụ tròn mà ta sẽ biểu diễn trên bản đồ bằng N hình tròn, hình tròn thứ i có tọa độ là (X i, Y i) và bán kính R i, các hình tròn không có điểm chung trong với nhau (nhưng có thể tiếp xúc).

Để đảm bảo an toàn phòng cháy chữa cháy, người ta cần xác định 2 bể chứa dầu gần nhau nhất để tăng cường cách ly khi xảy ra hỏa hoạn.

Biết rằng khoảng cách giữa 2 bể chứa dầu thứ i và thứ j chính bằng khoảng cách giữa 2 đường tròn tương ứng và bằng D ij– R i– R j, trong đó D ijlà khoảng cách Euclide giữa 2 điểm (X i, Y i) và (X j, Y j).

Bạn hãy giúp những người quản lý tìm ra 2 bể chứa dầu này.

Input

Dòng thứ nhất ghi số nguyên dương N là số bể chứa dầu.

Dòng thứ i trong N dòng tiếp theo ghi 3 số nguyên X i, Y i, R ilà tọa độ và bán kính bể chứa dầu thứ i.

Output

Gồm 1 dòng duy nhất là khoảng cách của 2 bể chứa dầu bé nhất tìm được.

Example

**Input:**

3

0 0 1

4 0 2

5 5 3

**Output:**

0.0990

**Giới hạn:**

2 ≤ N ≤ 10000.

|Xi|, |Yi| ≤ 106.

0 < Ri ≤ 106.

Kết quả ghi chính xác đến 4 chữ số sau dấu phẩy.

* **Người up:** cun
* **Điểm:** 1.31
* **Ngôn ngữ cho phép:**
* **Nguồn bài:** Ioicamp - Marathon 06 - 07

Sắp xếp các viên bi ( QBSORT )

**Giới hạn**

* **Thời gian:** 0.419s
* **Bộ nhớ:** 1536MB
* **Mã nguồn:** 50000 bytes

Có N viên bi màu được sắp thành một hàng trên mặt đất, các viên bi thuộc 1 trong K màu được đánh số từ 1 đến K.

Để tiện phân loại, beo\_chay\_so muốn sắp xếp lại các viên bi này sao cho các viên bi cùng màu thì nằm cạnh nhau, như vậy beo\_chay\_so sẽ thu được các đoạn liên tiếp gồm những viên bi cùng màu, mỗi màu chỉ thuộc đúng 1 đoạn.

Mỗi lần beo\_chay\_so chỉ được đổi chỗ 2 viên bi cạnh nhau, hãy giúp beo\_chay\_so sắp xếp lại các viên bi này sao cho số lần phải đổi chỗ các viên bi là ít nhất.

Input

Dòng thứ nhất ghi 2 số N và K là số viên bi và số màu. ( 2 ≤ N ≤ 20000, 1 ≤ K ≤ 10 )

Dòng thứ hai ghi N số nguyên dương là màu của N viên bi theo thứ tự.

Output

Ghi ra duy nhất một số nguyên là số phép đổi chỗ ít nhất.

Example

Input:

5 3

3 2 1 3 2

Output:

3

Giải thích:

Đổi chỗ số thứ 3 và 4:

3 2 3 1 2

Đổi chỗ số thứ 4 và 5:

3 2 3 2 1

Đổi chỗ số thứ 2 và 3:

3 3 2 2 1

* **Người up:** cun
* **Điểm:** 0.85
* **Ngôn ngữ cho phép:**

Tách từ ( WS )

**Giới hạn**

* **Thời gian:** 0.26s
* **Bộ nhớ:** 1536MB
* **Mã nguồn:** 50000 bytes

Một từ trong tiếng Việt được cấu tạo nên từ các âm tiết. Khi cho một câu trong tiếng Việt, việc tách các âm tiết thành các từ riêng biệt là một công việc cần thiết nhưng không đơn giản. Chẳng hạn câu: “Ông già đi nhanh quá” có thể được tách thành “/Ông/ già đi /nhanh / quá/” hoặc “Ông già/ đi / nhanh /quá”. Bạn là trợ lý của giáo sư có danh tiếng, công việc của bạn là giúp giáo sư viết một chương trình tách từ. Phần cốt lõi của chương trình là việc mô tả một cấu trúc dữ liệu cho phép thực hiện ba thao tác cơ bản trên 1 dãy n đơn âm cho trước (các đơn âm được đánh số lần lượt từ 1 tới n, theo thứ tự từ trái qua phải). Hai đơn âm cạnh nhau có thể được nối, hoặc không nối với nhau, một từ là một dãy đơn âm liên tiếp (cực đại) nối với nhau. Các thao tác trên cấu trúc dữ liệu bao gồm:

J i j: Nối từ đơn âm thứ i tới đơn âm thứ j với nhau

D i j: Tách các đơn âm từ đơn âm thứ i tới đơn âm thứ j (i ≤ j )

C: Đòi hỏi trả về số lượng từ ( số lượng dãy đơn âm nối nhau)

Chẳng hạn với n = 4 và trạng thái hiện tại của dãy đơn âm đang là ( 1\_2 3\_4 ) :

J 2 3 { Dãy đơn âm sẽ biến đổi thành: 1\_2\_3\_4 }

D 2 4 { Dãy đơn âm sẽ biến đổi thành: 1\_2 3 4 }

C { Trả về giá trị: 3 , do có ba từ là 1\_2, 3 và 4 }

Input

Dòng đầu ghi số K là số từ trong dãy ban đầu và số M là số lượng thao tác.

Dòng tiếp theo ghi K số nguyên dương lần lượt là độ dài của các từ tính từ trái qua phải

M dòng tiếp theo mỗi dòng bắt đầu bằng 1 chữ cái (J/D/C). Trong trường hợp chữ cái bắt đầu là J hoặc D tiếp theo ghi số hiệu từ mà thao tác cần thực hiện.

Output

Ứng với mỗi thao tác C trong input, ghi ra 1 số trên 1 dòng trong file output là số lượng từ tại thời điểm tương ứng.

Example

Input:

2 3

2 2

J 2 3

D 2 4

C

Output:

3

Giới hạn:

Số đơn âm không quá 50000.

M ≤ 100000

* **Người up:** cun
* **Điểm:** 0.56
* **Ngôn ngữ cho phép:**
* **Nguồn bài:** Ðinh Ngọc Thắng

Hệ quản trị cơ sở dữ liệu ( DBMS )

**Giới hạn**

* **Thời gian:** 0.28s
* **Bộ nhớ:** 1536MB
* **Mã nguồn:** 50000 bytes

Trong một hệ quản trị cơ sở dữ liệu thì một trong những khả năng quan trọng là điều khiển truy cập đồng thời, có nghĩa là khả năng xử lý các tình huống có nhiều người dùng cùng truy cập đến một vùng dữ liệu. Để làm được điều này thì một trong những cách phổ biến là khóa truy cập, có nghĩa là nếu một người đang làm việc với một vùng dữ liệu xác định thì những người khác muốn làm việc với vùng này đều phải chờ.

Cho một hệ cơ sở dữ liệu có N bản ghi được đánh số từ 1 đến N. Có M yêu cầu truy cập và cập nhật các bản ghi đối với hệ cơ sở dữ liệu này. Yêu cầu thứ i được biểu diễn bằng bộ 3 số nguyên dương (a i, b i, ti) có nghĩa là cần cập nhật các bản ghi từ thứ a iđến thứ b itại thời điểm t i. Danh sách các yêu cầu được cho là danh sách các yêu cầu được sắp xếp theo thời gian thực, nghĩa là thỏa mãn t i≤ t jvới mọi i < j.

Hệ quản trị cơ sở dữ liệu sẽ xử lý lần lượt M yêu cầu nói trên. Biết rằng:

- Để xử lý một yêu cầu nào đó thì hệ quản trị cần 1 đơn vị thời gian tuy nhiên nó có khả năng xử lý nhiều hơn 1 yêu cầu tại 1 thời điểm nếu các yêu cầu này không truy cập chung một bản ghi nào. Tại thời điểm này nó sẽ xử lý nhiều yêu cầu nhất có thể.

- Tại một thời điểm, nếu có nhiều hơn 1 yêu cầu cần xử lý thì hệ quản trị cơ sở dữ liệu sẽ xử lý theo thứ tự ưu tiên yêu cầu đứng trước trong danh sách các yêu cầu, nghĩa là yêu cầu có số hiệu nhỏ hơn.Ví dụ:

Các yêu cầu trong 1 thời điểm có số hiệu là 1, 2, ...i.Thì hệ sẽ xử lý yêu cầu 1 sau đó xét lần lượt các yêu cầu từ 2 đến i, gặp yêu cầu nào xử lý được thì xử lý và cứ tiếp tục như thế cho đến hết.

- Các yêu cầu chưa thể xử lý sẽ được cho vào hàng đợi để chờ xử lý tại thời điểm ngay sau đó. Nghĩa là tại thời điểm i chưa xử lý được sẽ chuyển sang xử lý ở thời điểm i+1.

Vì vậy một yêu cầu cần được xử lý tại thời điểm t thì có thể phải chờ đến thời điểm t’ mới được xử lý (t’ ≥ t), khi đó thời gian chờ đợi của yêu cầu này là t’ – t. Bạn hãy tính tổng thời gian chờ đợi của M yêu cầu.

Input

Dòng đầu ghi 2 số nguyên dương N và M.

Dòng thứ i trong M dòng tiếp theo ghi 3 số nguyên dương a i, b i, t i.

Output

Ghi ra duy nhất 1 số nguyên là tổng thời gian chờ đợi

Example

Input:

5 5

1 3 1

2 5 1

3 4 2

1 2 2

1 1 2

Output:

3

Giải thích ví dụ:

-Thời điểm 1: Các yêu cầu đem ra xử lý là 1, 2.

Xử lý yêu cầu 1, yêu cầu 2 chung 2 bản ghi 2, 3 với yêu cầu 1 nên chuyển sang xử lý ở thời điểm 2.

-Thời điểm 2: Các yêu cầu đem ra xử lý là 2, 3, 4, 5.

Xử lý yêu cầu 2, yêu cầu 2 và 3 có chung bản ghi với yêu cầu 2 nên chuyển sang thời điểm 3 để xử lý. Xử lý tiếp yêu cầu 5

-Thời điểm 3: Các yêu cầu đem ra xử lý là 3, 4.

Xử lý yêu cầu 3 và yêu cầu 4, vì chúng không có chung bản ghi nào

-Thời điểm xử lý các yêu cầu từ 1.. 5 tương ứng sẽ là 1, 2, 3, 3, 2

Giới hạn:

1 ≤ N ≤ 100000.

1 ≤ M ≤ 100000.

1 ≤ ai ≤ bi ≤ N.

1 ≤ ti ≤ 100000.

* **Người up:** cun
* **Điểm:** 1.86
* **Ngôn ngữ cho phép:**
* **Nguồn bài:** Dựa trên 1 bài của IOICAMP 3

**Trang 4**

Thăm quan công viên Disney ( QBDISNEY )

**Giới hạn**

* **Thời gian:** 0.059s
* **Bộ nhớ:** 1536MB
* **Mã nguồn:** 50000 bytes

Công viên DISNEYLAND là một công viên hiện đại mới được xây dựng ở ngoại ô Hà Nội để dành riêng cho trẻ em. Công viên này có N điểm vui chơi được đánh số từ 1 đến N. Các điểm vui chơi được nối liền với nhau bằng các đoạn đường hai chiều theo hướng Bắc – Nam, hoặc Đông – Tây. Các điểm vui chơi nằm ở giao của hai con đường. Khoảng cách giữa hai điểm vui chơi liên tiếp cách đều nhau.

Bé Bi được bố mẹ cho phép đến công viên chơi trong một ngày. Vì công viên quá rộng lớn và hấp dẫn nên bé Bi muốn lựa chọn một con đường đi qua một số điểm vui chơi sao cho không có điểm vui chơi nào đi qua 2 lần. Vì bố mẹ sợ bé Bi bị lạc, nên bố mẹ chỉ cho phép bé đi trên đoạn đường gồm không quá K ngã rẽ (không được đổi hướng quá K lần). Biết rằng độ dài đoạn đường nối hai điểm kề nhau có giá trị là 1. Bạn hãy giúp bố mẹ bé tính xem, đoạn đường dài nhất thỏa mãn điều kiện đó có độ dài là bao nhiêu, và có bao nhiêu đoạn đường thỏa mãn điều kiện đó. Lưu ý rằng công viên rất hiện đại nên có cả hệ thống cầu vượt và hầm chui. Chính vì vậy sẽ có những trường hợp như test ví dụ thứ hai dưới đây (các đỉnh ở cùng tọa độ nhưng hoàn toàn phân biệt với nhau).

Input

Dòng thứ nhất ghi 2 số nguyên dương N và K là số điểm vui chơi và số ngã rẽ nhiều nhất. ( 0 ≤ K < N ≤ 10000 )

Dòng thứ i trong N dòng tiếp theo ghi 4 số nguyên mô tả thông tin về điểm vui chơi thứ i. Bốn số nguyên là L, D, R, U là điểm vui chơi nằm bên trái, dưới, phải, trên điểm vui chơi thứ i. (Nếu có số bằng 0 tương ứng với phía đó không có điểm vui chơi nào).

Output

Gồm 2 số nguyên S và P trong đó S là độ dài đường đi dài nhất và P là số đường đi có độ dài là S.

Example

Input:

12 4

0 2 3 4

0 0 0 1

1 0 0 0

5 1 0 0

6 0 4 0

0 7 5 0

8 0 9 6

10 11 7 12

7 0 0 0

0 0 8 0

0 0 0 8

0 8 0 0

Output:

7 4

Input:

5 2

0 0 2 0

1 3 0 0

4 0 0 2

0 0 3 5

0 4 0 0

Output:

3 2

* **Người up:** cun
* **Điểm:** 1.60
* **Ngôn ngữ cho phép:**

Congruence Equation ( DPEQN )

**Giới hạn**

* **Thời gian:** 1.0s
* **Bộ nhớ:** 1536MB
* **Mã nguồn:** 50000 bytes

Cho phương trình đồng dư:

a1x1 + ... a2x2 + ... + anxn = b (mod m)

Trong đó a 1, a 2, ..., a n, b và m là các hằng số nguyên dương cho trước; x 1, x 2, ..., x nlà các ẩn.

Tìm một nghiệm của phương trình trên, hoặc thông báo phương trình vô nghiệm.

Dữ liệu

Dòng đầu tiên ghi số bộ test, mỗi bộ test có dạng như sau:

* Dòng 1: n (1 ≤ n ≤ 100)
* Dòng 2: gồm n số nguyên a 1, a 2, ..., a n(1 ≤ a i≤ 10 8)
* Dòng 3: b, m (1 ≤ b, m ≤ 10 8)

Mỗi bộ test được phân cách bởi một dòng trắng ở đầu.

Kết quả

Với mỗi bộ test, nếu phương trình không có nghiệm , in ra dòng "NO". Trong trường hợp có nghiệm, in ra trên một dòng n số nguyên x 1, x 2, ..., x n(0 ≤ x i< m) là một nghiệm tìm được.

Ví dụ

Dữ liệu

2

2

4 6

6 10

2

4 6

3 8

Kết quả

1 2

NO

* **Người up:** paulmcvn
* **Điểm:** 1.40
* **Ngôn ngữ cho phép:**
* **Nguồn bài:** © VNOI

Divisibility Relation ( DIVREL )

**Giới hạn**

* **Thời gian:** 0.194s
* **Bộ nhớ:** 1536MB
* **Mã nguồn:** 50000 bytes

Quan hệ chia hết

Cho n số nguyên dương. Hãy chọn ra nhiều số nhất sao cho không có hai số a, b nào được chọn mà a chia hết cho b.

Dữ liệu

* Dòng 1: n (1 ≤ n ≤ 200).
* Dòng 2: n số nguyên dương a 1, a 2, ..., a n(1 ≤ a i≤ 10 9).

Kết quả

* Dòng 1: k, số số nhiều nhất có thể chọn.
* Dòng 2: k số được chọn.

Ví dụ

Dữ liệu

8

1 2 3 5 6 8 7 9

Kết quả

5

5 6 8 7 9

Dữ liệu

4

2 3 2 3

Kết quả

2

2 3

* **Người up:** paulmcvn
* **Điểm:** 0.75
* **Ngôn ngữ cho phép:**
* **Nguồn bài:** Vietnamese IOI Selection Test 2007

Tree Construction ( TREECST )

**Giới hạn**

* **Thời gian:** 3.0s
* **Bộ nhớ:** 1536MB
* **Mã nguồn:** 50000 bytes

Xây dựng cây

Cho một cây có N đỉnh. Tìm cách xóa đi một cạnh thuộc cây và thêm vào một cạnh mới, sao cho sau đó, độ dài đường đi dài nhất trên cây là nhỏ nhất có thể. Độ dài của một đường đi được tính bằng số cạnh thuộc đường đi đó.

Dữ liệu

Dòng đầu tiên chứa số N (1 ≤ N ≤ 300 000).

N-1 dòng sau, mỗi dòng chứa 2 số nguyên mô tả một cạnh của cây.

Kết quả

* Dòng đầu tiên in ra độ dài nhỏ nhất tìm được.
* Dòng thứ hai ghi 2 số nguyên cho biết cạnh cần xóa.
* Dòng thứ ba ghi 2 số nguyên cho biết cạnh cần thêm vào.

Nếu có nhiều lời giải, chỉ cần in ra một lời giải bất kỳ.Ví dụ

Dữ liệu

4

1 2

2 3

3 4

Kết quả

2

3 4

4 2

Dữ liệu

7

1 3

2 3

2 7

4 3

7 5

3 6

Kết quả

3

2 3

7 3

* **Người up:** paulmcvn
* **Điểm:** 1.16
* **Ngôn ngữ cho phép:**
* **Nguồn bài:** COCI 2008-2009, #1

Kiềng ba chân ( TRIPOD )

**Giới hạn**

* **Thời gian:** 0.079s
* **Bộ nhớ:** 1536MB
* **Mã nguồn:** 50000 bytes

Trong một chuyến đi dã ngoại, BB muốn cùng các bạn bắc bếp nấu cơm trên khu đất cắm trại. Trên mặt đất có n hòn đá, không có 2 hòn đá nào có cùng vị trí. Mọi người sẽ chọn ra 3 trong số các hòn đá này để bắc nồi lên. 3 hòn đá phải được chọn sao cho bán kính hình tròn chứa chúng là nhỏ nhất vì nếu có một cái nồi nhỏ bắc được lên 3 hòn đá này thì cái nồi có bán kính lớn hơn cũng có thể bắc được lên 3 hòn đá này. Bạn hãy giúp BB tìm ra 3 hòn đá cần chọn.

Input

Dòng đầu tiên ghi số n là số hòn đá.

Trong n dòng sau, dòng thứ u ghi một cặp số nguyên x u, y ulà tọa độ của hòn đá thứ u.

Output

Ghi ra bán kính hình tròn chứa 3 hòn đá tìm được ( Chính xác đến 5 chữ số sau dấu phẩy ).

Example

Input:

3

0 0

4 0

0 4

Output:

2.82842

Giới hạn:

3 ≤ N ≤ 1000.

|xu|, |yu| ≤ 30000

* **Người up:** cun
* **Điểm:** 1.05
* **Ngôn ngữ cho phép:**
* **Nguồn bài:** Ioicamp - marathon 05 - 06