



Cho một tập hợp hoặc dãy số  $S$ ,  $mex(S)$  là số nguyên không âm nhỏ nhất không có trong  $S$ . Ví dụ:  $mex(\{4, 33, 0, 1, 1, 5\}) = 2$ ,  $mex(\{1, 2, 3\}) = 0$ .

Cho mảng  $a[]$  gồm  $N$  số nguyên không âm và  $M$  truy vấn. Mỗi truy vấn được cho bởi một số nguyên  $x$ , bạn cần thực hiện tuần tự các bước sau:

1. Cập nhật lại mảng  $a[]$ : Thay thế mỗi phần tử  $a_i$  trong mảng bằng kết quả của phép toán XOR theo bit giữa  $a_i$  và  $x$  ( $a_i = a_i \text{ XOR } x$ ).
2. In ra giá trị  $mex$  của mảng  $a[]$  sau khi đã cập nhật.

Lưu ý rằng  $a[]$  bị thay đổi sau mỗi truy vấn.

Input:

Dòng đầu tiên gồm hai số nguyên  $N$  và  $M$  ( $1 \leq N, M \leq 300000$ ).

Dòng thứ hai gồm  $N$  số nguyên  $a_i$  ( $0 \leq a_i \leq 300000$ ).

$M$  dòng tiếp theo, mỗi dòng chứa một số nguyên  $x$  ( $0 \leq x \leq 300000$ ).

Output

Với mỗi truy vấn, in ra giá trị  $mex$  của mảng sau khi cập nhật.

Test ví dụ:

Input	Output
2 2 1 3 1 3	1 0
4 3 0 1 5 6 1 2 4	2 0 0
5 5 1 2 3 4 5 1 2 3 4 5	1 3 0 2 1



## BÀI TẬP

Dãy ngoặc đúng được định nghĩa như sau:

- Một xâu rỗng được coi là một dãy ngoặc đúng.
- Nếu A và B là dãy ngoặc đúng thì xâu AB cũng là một dãy ngoặc đúng.
- Nếu A là dãy ngoặc đúng thì (A) cũng là một dãy ngoặc đúng.

Cho xâu S chứa các ký tự (, ), và một ký tự \*.

Nhiệm vụ của bạn là hãy đếm xem có bao nhiêu cặp dấu ngoặc tương ứng xung quanh ký tự \*. Cặp ký tự  $S_L$  và  $S_R$  được gọi là cặp dấu ngoặc tương ứng nếu như  $S_L$  là ký tự mở ngoặc,  $S_R$  là ký tự đóng ngoặc và xâu con từ vị trí  $S_L$  đến  $S_R$  khi bỏ ký tự \* đi là một dãy ngoặc đúng.

Input

Dòng đầu tiên là số lượng bộ test T ( $T \leq 100$ ).

Mỗi test gồm một xâu S có không quá 1000 ký tự.

Output

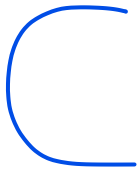
Với mỗi test, hãy in ra kết quả tìm được trên một dòng.

Test ví dụ:

Input	Output
4	2
((*)())	1
(*)	0
((())*)	0
()*()	

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 524288 Kb



## BÀI TẬP

Cho  $N$  số nguyên  $A[1], A[2], \dots, A[N]$  thỏa mãn  $A[1] \times A[2] \times \dots \times A[N] = P$  với số  $P$  cho trước.

Nhiệm vụ của bạn là hãy tính giá trị lớn nhất có thể của  $\text{GCD}(A[1], A[2], \dots, A[N])$ .

**Input:**

Input gồm hai số nguyên dương  $N$  và  $P$  ( $1 \leq N, P \leq 10^{12}$ ).

**Output:**

In ra một số nguyên là đáp án tìm được.

**Test ví dụ:**

Input:	Output:
3 24	2
3 1	1
1 20	20

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 524288 Kb



## BÀI TẬP

Có  $N$  lá bài, mỗi lá bài có gắn với một con số nguyên dương  $A[i]$ . Bạn có thể chọn ngẫu nhiên ba lá bài rồi ghép chúng với nhau theo thứ tự tùy ý để tạo thành một con số mới.

Ví dụ như ba lá bài 1, 23, 4, bạn có thể tạo thành số 1234 hay 2314.

Hãy tìm số nguyên lớn nhất có thể tạo ra?

Input:

Dòng đầu tiên là số nguyên dương  $N$  ( $3 \leq N \leq 200000$ ).

Dòng thứ hai nhập  $N$  số nguyên  $A[1], A[2], \dots, A[N]$  ( $1 \leq A[i] \leq 10^6$ ).

Output:

In ra giá trị số nguyên lớn nhất tìm được.

Test ví dụ:

Input	Output
6 23 54 54 98 90 11	989054
5 1 2 3 4 5	543

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 524288 Kb

## BÀI TẬP

Cho 2 cây, mỗi cây có  $N-1$  cạnh. Một cây có các cạnh được tô màu xanh, còn một cây có các cạnh được tô màu đỏ. Nhiệm vụ của bạn là hãy biến đổi cây màu xanh sang cây màu đỏ bằng cách sử dụng thao tác sau đúng  $N-1$  lần:

- Chọn một đường đi từ  $x \rightarrow y$  chứa toàn các cạnh màu xanh trên cây thứ nhất, sau đó được phép xóa bỏ 1 cạnh trên đường đi này.
- Tạo cạnh mới  $x-y$  màu đỏ với  $x, y$  là điểm đầu và điểm cuối của đường đi ở trên.

Input:

Dòng đầu tiên là số nguyên dương  $N$  ( $2 \leq N \leq 100000$ ).

$N-1$  dòng tiếp theo, mỗi dòng gồm 2 số nguyên mô tả một cạnh của cây thứ nhất.

$N-1$  dòng còn lại, mỗi dòng gồm 2 số nguyên mô tả một cạnh của cây thứ hai.

Output:

In ra "YES" nếu như có thể biến đổi được cây thứ nhất sang thành cây thứ hai, in ra "NO" trong trường hợp ngược lại.

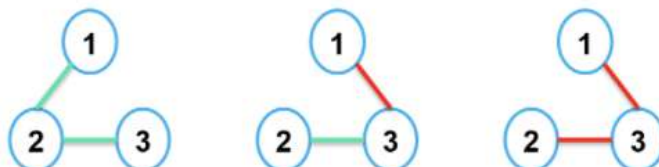
Test ví dụ:

Input	Output
3 1 2 2 3 1 3 3 2	YES
5 1 2 2 3 3 4 4 5 1 4 1 5 2 4 3 4	YES
6 1 2 3 5 1 6 4 6 5 1 1 4 2 6 5 3 4 3 5 6	NO

Giải thích test 1:

B1: Chọn đường đi (1, 3), xóa cạnh  $1 \rightarrow 2$  và tô màu đỏ cho cạnh  $1 \rightarrow 3$ .

B2: Chọn đường đi (2, 3), xóa chính cạnh này rồi tô bằng màu đỏ.





## BÀI TẬP

Cho mảng  $A[]$  gồm  $N$  phần tử, ban đầu tất cả  $A_i = 1$ . Bạn được phép thực hiện tối đa  $K$  thao tác, mỗi thao tác như sau:

- Chọn một chỉ số  $i$  ( $1 \leq i \leq N$ ) và một số nguyên  $x > 0$ .
- Cập nhật  $A_i = A_i + \lfloor A_i/x \rfloor$  với  $\lfloor \cdot \rfloor$  là kí hiệu của phần nguyên.

Sau khi kết thúc chuỗi các thao tác, bạn sẽ nhận được  $C_i$  xu cho mỗi cặp giá trị  $A_i = B_i$ . Hãy tính số đồng xu tối đa bạn có thể nhận được.

Input:

Dòng đầu tiên là số lượng bộ test  $T$  ( $T \leq 100$ ).

Dòng đầu tiên của mỗi test gồm số nguyên  $N$  và  $K$  ( $1 \leq N \leq 1000$ ,  $0 \leq K \leq 1000000$ ).

Dòng thứ hai gồm  $N$  số nguyên  $B_1, B_2, \dots, B_N$  ( $1 \leq B_i \leq 1000$ ).

Dòng thứ hai gồm  $N$  số nguyên  $C_1, C_2, \dots, C_N$  ( $1 \leq C_i \leq 1000000$ ).

Output:

Với mỗi test, hãy in ra số lượng đồng xu tối đa bạn nhận được.

Test ví dụ:

Input	Output
4	7
4 4	0
2 7 5 2	30
3 6 4 2	159
3 0	
3 5 2	
5 4 9	
5 9	
5 2 5 6 3	
5 9 1 9 7	
6 14	
12 5 6 4 8 15	
44 35 6 41 18 39	

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 500000 KB



## BÀI TẬP

Một chuỗi được gọi là chuỗi lặp lại hay chuỗi Double nếu như nó độ dài chẵn và nửa trái của chuỗi giống với nửa bên phải còn lại. Ví dụ như abcabc, dd, efef là các chuỗi Double.

Cho một chuỗi S cho trước, nhiệm vụ của bạn là hãy đếm số cách chọn ra 2 chuỗi con liên tiếp riêng biệt và khác rỗng để có thể ghép lại thành một chuỗi Double? (Thứ tự 2 chuỗi con vẫn giữ nguyên theo vị trí xuất hiện của chúng trong chuỗi S ban đầu).

Ví dụ như với S = "aaaa", chúng ta có 9 cách chọn:

Tạo ra chuỗi Double "aa": có 6 cách với các tổ hợp chọn ký tự (1, 2), (1, 3), (1, 4), (2, 3), (2, 4), (3, 4).

Tạo ra chuỗi Double "aaaa" có 3 cách: "a" + "aaaa", "aa" + "aa", "aaa" + "a".

Input:

Dữ liệu đầu vào chứa một chuỗi S có độ dài không vượt quá 1000 ký tự, chỉ gồm các chữ cái thường.

Output:

In ra một số nguyên là số cách ghép tạo ra chuỗi Double tìm được.

Test ví dụ:

Input:	Output:
aaaa	9
aeacebdebdeb	32

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 524288 Kb

H

## BÀI TẬP

Cho bảng số nguyên không âm kích thước  $M \times N$  và hai số lẻ  $r, c$ . Hãy tìm bảng con kích thước  $r \times c$  để giá trị trung vị của các số trong bảng con là nhỏ nhất.

### Input

Dòng đầu tiên gồm 4 số nguyên  $M, N, r, c$  ( $1 \leq M, N \leq 1000$ ).

$M$  dòng tiếp theo, mỗi dòng gồm  $N$  số nguyên không âm mô tả bảng số, các số không vượt quá  $10^9$ .

### Output

In ra giá trị phần tử trung vị nhỏ nhất tìm được.

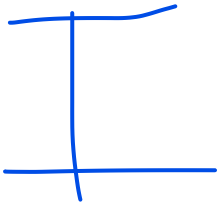
Test ví dụ:

Input	Output
5 5 3 3 5 11 12 16 25 17 18 2 7 10 4 23 20 3 1 24 21 19 14 9 6 22 8 13 15	9

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 524288 Kb





## BÀI TẬP

Trên mặt phẳng Oxy, cho  $N$  điểm  $P_i$ . Tọa độ của giữa 2 điểm  $(x_1, y_1)$  và  $(x_2, y_2)$  được tính theo khoảng cách Euclidean:  $\sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$ .

Với mỗi điểm, bạn hãy tìm chỉ số ID của điểm xa nhất với nó.

### Input

Dòng đầu tiên là số nguyên dương  $N$  ( $2 \leq N \leq 100$ ).

$N$  dòng tiếp theo, mỗi dòng gồm 2 số nguyên  $x_i, y_i$  mô tả tọa độ của một điểm. Input đảm bảo tọa độ các điểm là riêng biệt. Có tọa độ có giá trị tuyệt đối không vượt quá 1000.

### Output

In ra  $N$  dòng, dòng thứ  $i$  là ID của điểm cách xa điểm  $P_i$  nhất. Nếu có nhiều đáp án, hãy in ra điểm có ID nhỏ hơn.

Test ví dụ:

Input	Output
4	3
0 0	3
2 4	1
5 0	1
3 4	
5	5
1 1	5
2 2	1
3 3	1
4 4	1
5 5	

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 524288 Kb



## BÀI TẬP

Đất nước Byteland có  $N$  thành phố và  $M$  tuyến đường hai chiều. Tuyến đường thứ  $i$  kết nối thành phố  $A[i]$  với  $B[i]$ . Tại các mốc thời gian  $0, K[i], 2 \cdot K[i], \dots$  (mốc thời gian là bội của  $K[i]$ ) sẽ có một chuyến tàu đi chuyển từ thành phố  $A[i]$  tới  $B[i]$  và một chuyến tàu cho chiều ngược lại. Thời gian để mỗi chuyến tàu có thể tới đích là  $T[i]$ .

Hiện tại, bạn đang ở thành phố  $X$ . Hãy tính xem thời gian sớm nhất bạn có thể tới được thành phố  $Y$  là bao nhiêu? Thời gian chuyển tiếp giữa các chuyến tàu có thể bỏ qua, tức là bạn có thể lên một chuyến tàu mới ngay lập tức sau khi vừa kết thúc một chuyến tàu cũ.

### Input:

Dòng đầu tiên chứa 4 số nguyên  $N, M, X$  và  $Y$  ( $2 \leq N \leq 100000, 0 \leq M \leq 100000, 1 \leq X, Y \leq N$ ).

$M$  dòng tiếp theo, mỗi dòng gồm 4 số nguyên  $A[i], B[i], T[i]$  và  $K[i]$  ( $A[i]$  khác  $B[i], 1 \leq T[i], K[i] \leq 10^9$ ).

### Output:

In ra một số nguyên là thời gian sớm nhất bạn có thể tới được thành phố  $Y$ . Nếu không tồn tại đáp án, hãy in ra  $-1$ .

### Test ví dụ:

Input	Output
3 2 1 3 1 2 2 4 2 3 4 4	8
3 2 3 1 1 2 2 3 2 3 3 4	5
3 0 3 1	-1

Giải thích test 1: Lên tàu tại mốc thời gian  $t = 0$  từ thành phố 1 tới thành phố 2 tại  $t = 2$ , đợi tới thời điểm  $t = 4$ , bạn lên chuyến tàu từ thành phố 2 tới thành phố 3, tới nơi tại thời điểm  $t = 8$ .

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 524288 Kb