1001 The Game

Description

John and his brother are playing a game. There are several piles of stones. At first John has to pick up some stones from one pile. Then his brother has to make a turn. And so on. Everyone has to pick up at least one stone during his turn. If John (or his brother) will pick up the last stone, he will be considered as a looser.
Both of players are using optimal game strategy. John starts first always.

Input

The first line of input contains a single integer **T (1<= T <= 400)** indicating the number of test cases. Each test case starts with number **N** of piles **(1<= N <= 47)**. Then the next line will contain **N** integers $\mathbf{S_i}$ **(1<=** $\mathbf{S_i}$ **<= 4700)**, separated by spaces – amount of stones of **i**-th pile.

Output

Output T lines each of them containing information about game winner. Print "John" if John will win the game, or "Brother" in other case.

Sample Input

2
3
3 5 1
1
1

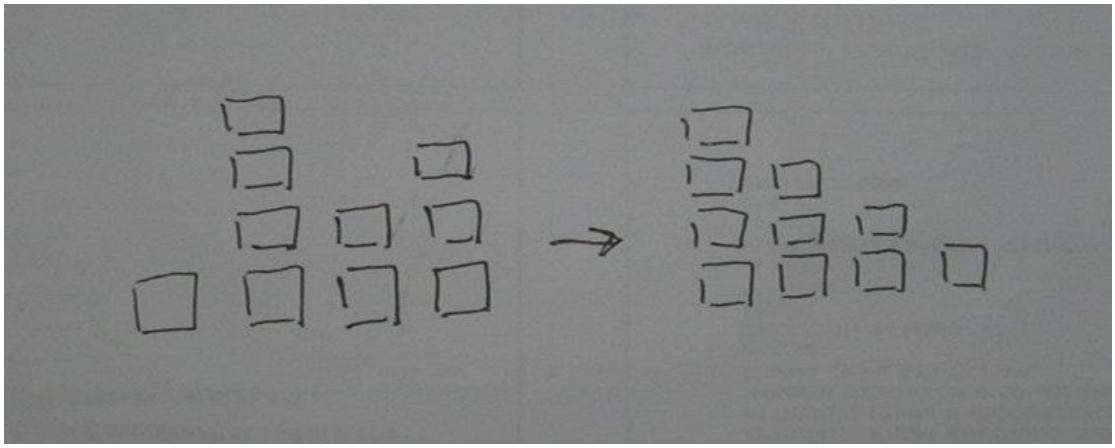
Sample Output

John
Brother

1002 Sherlock I

Description

Sherlock Holmes is an excellent detective and Moriarty is the big villain. One day Moriarty kidnaps Sherlock's boyfriend ---- Jhon Watson. Of course,Sherlock goes to save him. As Moriarty likes to play the game 2048, he makes a lock as the game on the door.
The lock has N columns, each column has some diamonds, when Sherlock touches the lock, all diamonds slide to the left if they can,the diamonds won't combine ， the new sequence is the password.



Input

There are several cases. For each case, the first line is N(n<=100000), and then n integers(0<=each integer<=1000)

Output

Output the new sequence in a line.

Sample Input

4
1 4 2 3

Sample output

4 3 2 1

1003 Sherlock II

Description

Sherlock Holmes is an excellent detective and Moriarty is the big villain. One day Moriarty kidnaps Sherlock's boyfriend ---- Jhon Watson. Of course,Sherlock goes to save him. As sherlock has found the password in the previous problem.When he finds Jhon, there is a time bomb. Sherlock has to defuse the bomb.
The bomb has a special coded lock,the lock has two sequence A and B, Sherlock has to find a new Sequence C , C is both the subsequence of A and B, and for each i<j C[i]<C[j]. The length of C should be as long as possible.The length of sequence C is the password.

Input

There are several cases.
For each case,the first line is N and M (1<=n,m<=100)
And the is sequence A and B , length(A)=N length(B)=M （1<= A[i],B[i]<=50）

Output

The length of sequence C.

Sample input

5 5
1 2 0 4 5
1 0 4 5 2

Sample output

3

1004 XL's Math Problem I

Time limit : 10s

Calculate $S_n = \sum_{i=1}^{n} i^k (n < 10^9, k < 100)$

output $S_n \mod 10^9 + 7$

Input

First line is the number of cases and then n and k in each case.

Sample Input:

5

10 1

10 2

Sample Output:

Case1: 55

Case2: 385

1005 XL's Math Problem II

Time limit : 5s

calculate $S_n = \sum_{i=1}^{n} \frac{n}{i} (n < 10^9)$

Input

First line is the number of cases and then n in each case.

Sample Input:

5
6
7
8
9
10

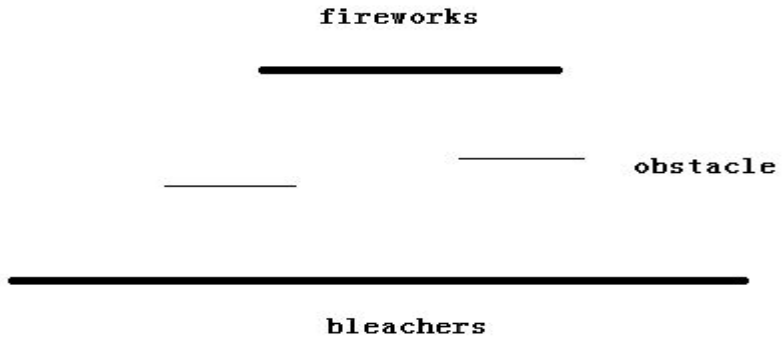
Sample Output:

Case 1: 14
Case 2: 16
Case 3: 20
Case 4: 23
Case 5: 27

1006 Fireworks

Description

There are a row of fireworks that parallel to the X-axis and a bleachers on the X-axis. There are some obstacles that also parallel to the X-axis between the fireworks ant the bleachers.



Input

There are several cases.
Fireworks, bleachers and obstacles all can be regarded as segments that parallel to the X-axis. They have 3 parameters x1,x2 and y(x1,x2,y are integers<=100000). Y of the bleachers is always 0 , Y of fireworks must greater than 0 and Y of obstacles between them.
For each case, line 1 and line 2 are the parameters of fireworks and the bleachers, and then is N（n<=40）,means N obstacles. Line 3 to line n+2 describe the parameters of the obstacles.

Output

Output the longest distance on the bleachers that can see all the fireworks, keep 2 decimal places.
Output “0.00” if can’t see the fireworks.

Sample Input

0 1 2
0 3 0
1
0 1 1

Sample Output

1.00

1007 Interval Mode

Description

Given a non-decreasing ordered sequence of n integers $\mathbf{S_1, S_2, \dots, S_n}$, and two integers **a** and **b**, we can make the definition of **interval mode**: the most frequent integer $\mathbf{S_i}$ among the integers $\mathbf{S_a, \dots, S_b}$ **(a <= i <= b)**. Getting the interval mode of the interval **[a, b]** is a little bit complex, so your task is to find the **number of occurrences** of the interval mode within the interval.

Input

The input consists of multiple datasets. Each starts with a line containing two integers **n** and **q** (1 <= n, q <= 100000). The next line contains **n** integers $\mathbf{S_1, \dots, S_n}$ (-100000 <= S_i <= 100000, 1<= i <= n), and the adjacent integers are separated by spaces. You can assume that $\mathbf{S_i <= S_{i+1}}$ (1 <= i <= n). The following q lines contain one query each, consisting of two integers a and b (1 <= a <= b <= n), which indicate the boundary of the interval.
The last test case only contains a single 0 and should not be processed.

Output

For each query, print one line with one integer: The number of occurrences of the interval mode within the interval.

Sample Input

```
10 4
-2 -2 3 3 3 3 5 6 6 6
2 3
1 10
1 5
1 3
0
```

Sample Output

```
1
4
3
2
```

1008 Repair Streets

Description

A company have Q streets to repair, numbered from 1 to Q. One day the get M tasks , each task has 3 parameters, A, B and C. (A is the id of the street, B is the time to start and C is the time to finish). Each task need one worker to do,and the worker can do another task after finishing what he is doing. Of course, there may be some tasks at the same street at the same time. Can you figure out how many worker does the company need at least?

Input

There are several cases. For each case, the first line is Q and M （0<Q<=20,0<M<=200）, means the number of streets and the tasks, and then a Q*Q array T, T[i][j] means the minimal time from street i to street j (it is already the minimal time).0<=T[i][j]<=100000
And then Q lines follows, each line describes a task. a<=Q

Output

The number of workers at least.

Sample Input

1 2
0
1 1 7
1 5 10

Sample Output

2

1009 The last one maybe the easiest

Description

As the title, you can solve this problem quickly. You have a sequence A ={ 1,2,4,8,16.....2^50 },and you can choose some numbers and add them up. We give you some numbers, can you tell us how to choose numbers in sequence A?

Input

There are several cases.
For each case contains an integer N (1<=N<=2^50).

Output

Output the numbers you choose in increasing order.

Sample input

6
5

Sample output

2 4
1 4