A. Maximize?

Input file: standard input
Output file: standard output
Time limit: 2 seconds
Memory limit: 256 megabytes

You are given an integer x. Your task is to find any integer y $(1 \le y < x)$ such that $\gcd(x, y) + y$ is maximum possible.

Note that if there is more than one y which satisfies the statement, you are allowed to find any.

$\gcd(a, b)$ is the Greatest Common Divisor of a and b. For example, $\gcd(6, 4) = 2$.

Input

The first line contains a single integer t $(1 \le t \le 1000)$ — the number of test cases.

Each of the following t lines contains a single integer x $(2 \le x \le 1000)$.

Output

For each test case, output any y $(1 \le y < x)$, which satisfies the statement.

Standard Input	Standard Output
7 10 7 21 100 2 1000 6	5 6 18 98 1 750 3

,

B. Prefiquence

```
Input file:     standard input
Output file:    standard output
Time limit:     2 seconds
Memory limit:   256 megabytes
```

You are given two binary strings a and b. A binary string is a string consisting of the characters '0' and '1'.

Your task is to determine the maximum possible number k such that a prefix of string a of length k is a subsequence of string b.

A sequence a is a subsequence of a sequence b if a can be obtained from b by the deletion of several (possibly, zero or all) elements.

Input

The first line consists of a single integer t ($1 \le t \le 10^4$) — the number of test cases.

The first line of each test case contains two integers n and m ($1 \le n, m \le 2 \cdot 10^5$) — the length of string a and the length of string b, respectively.

The second line of each test case contains a binary string a of length n.

The third line of each test case contains a binary string b of length m.

It is guaranteed that the sum of values n over all test cases does not exceed $2 \cdot 10^5$. Similarly, the sum of values m over all test cases does not exceed $2 \cdot 10^5$.

Output

For each test case, output a single number — the maximum k, such that the first k characters of a form a subsequence of b.

Standard Input	Standard Output
6 5 4 10011 1110 3 3 100 110 1 3 1 111 4 4 1011 1111 3 5 100 11010 3 1 100 0	2 2 1 1 3 0

Note

In the first example, the string '10' is a subsequence of '1110' but the string '100' is not. So the answer is 2.

In the fifth example, a='100', b='$1\color{red}{1}0\color{red}{1}0$', whole string a is a subsequence of string b. So the answer is 3.

In the sixth example, string b does not contain '1' so the answer is 0.

,

C. Assembly via Remainders

Input file: standard input
Output file: standard output
Time limit: 2 seconds
Memory limit: 256 megabytes

You are given an array $x_2, x_3, \ldots, x_n$. Your task is to find **any** array $a_1, \ldots, a_n$, where:

- $1 \le a_i \le 10^9$ for all $1 \le i \le n$.
- $x_i = a_i \bmod a_{i-1}$ for all $2 \le i \le n$.

Here $c \bmod d$ denotes the remainder of the division of the integer c by the integer d. For example $5 \bmod 2 = 1$, $72 \bmod 3 = 0$, $143 \bmod 14 = 3$.

Note that if there is more than one a which satisfies the statement, you are allowed to find any.

Input

The first line contains a single integer t $(1 \le t \le 10^4)$ — the number of test cases.

The first line of each test case contains a single integer n $(2 \le n \le 500)$ — the number of elements in a.

The second line of each test case contains $n - 1$ integers $x_2, \ldots, x_n$ $(1 \le x_i \le 500)$ — the elements of x.

It is guaranteed that the sum of values n over all test cases does not exceed $2 \cdot 10^5$.

Output

For each test case output any $a_1, \ldots, a_n$ $(1 \le a_i \le 10^9)$ which satisfies the statement.

Standard Input	Standard Output
5 4 2 4 1 3 1 1 6 4 2 5 1 2 2 500 3 1 5	3 5 4 9 2 5 11 5 14 16 5 11 24 501 500 2 7 5

Note

In the first test case $a = [3, 5, 4, 9]$ satisfies the conditions, because:

- $a_2 \bmod a_1 = 5 \bmod 3 = 2 = x_2$;
- $a_3 \bmod a_2 = 4 \bmod 5 = 4 = x_3$;
- $a_4 \bmod a_3 = 9 \bmod 4 = 1 = x_4$;

,

D. Permutation Game

```
Input file:     standard input
Output file:    standard output
Time limit:     2 seconds
Memory limit:   256 megabytes
```

Bodya and Sasha found a permutation $p_1, \ldots, p_n$ and an array $a_1, \ldots, a_n$. They decided to play a well-known "Permutation game".

A permutation of length n is an array consisting of n distinct integers from 1 to n in arbitrary order. For example, $[2, 3, 1, 5, 4]$ is a permutation, but $[1, 2, 2]$ is not a permutation (2 appears twice in the array), and $[1, 3, 4]$ is also not a permutation ($n = 3$ but there is 4 in the array).

Both of them chose a starting position in the permutation.

The game lasts k turns. The players make moves simultaneously. On each turn, two things happen to each player:

- If the current position of the player is x, his score increases by a_x.
- Then the player either **stays** at his current position x or **moves** from x to p_x.

The winner of the game is the player with the higher score after exactly k turns.
Knowing Bodya's starting position P_B and Sasha's starting position P_S, determine who wins the game if both players are trying to win.

Input

The first line contains a single integer t ($1 \le t \le 10^4$) — the number of testcases.

The first line of each testcase contains integers n, k, P_B, P_S ($1 \le P_B, P_S \le n \le 2 \cdot 10^5$, $1 \le k \le 10^9$) — length of the permutation, duration of the game, starting positions respectively.

The next line contains n integers $p_1, \ldots, p_n$ ($1 \le p_i \le n$) — elements of the permutation p.

The next line contains n integers $a_1, \ldots, a_n$ ($1 \le a_i \le 10^9$) — elements of array a.

It is guaranteed that the sum of values of n over all test cases does not exceed $2 \cdot 10^5$.

Output

For each testcase output:

- "`Bodya`" if Bodya wins the game.
- "`Sasha`" if Sasha wins the game.
- "`Draw`" if the players have the same score.

Standard Input	Standard Output
10 4 2 3 2 4 1 2 3 7 2 5 6 10 8 2 10 3 1 4 5 2 7 8 10 6 9 5 10 5 1 3 7 10 15 4 3 2 1000000000 1 2 1 2 4 4 8 10 4 1	Bodya Sasha Draw Draw Bodya Sasha Sasha Sasha Sasha Bodya

5 1 4 3 2 8 6 7 1 1 2 1 2 100 101 102 5 1 2 5 1 2 4 5 3 4 6 9 4 2 4 2 3 1 4 1 3 2 6 8 5 3 6 9 5 4 6 1 3 5 2 4 6 9 8 9 5 10 4 8 4 2 2 3 4 1 5 2 8 7 4 2 3 1 4 1 3 2 6 8 5 3 2 1000000000 1 2 1 2 1000000000 2	

Note

Below you can find the explanation for the first testcase, where the game consists of $k = 2$ turns.

Turn	Bodya's position	Bodya's score	Bodya's move	Sasha's position	Sasha's score	Sasha's move
first	3	$0 + a_3 = 0 + 5 = 5$	stays on the same position	2	$0 + a_2 = 0 + 2 = 2$	moves to $p_2 = 1$
second	3	$5 + a_3 = 5 + 5 = 10$	stays on the same position	1	$2 + a_1 = 2 + 7 = 9$	stays on the same position
final results	3	10		1	9	

As we may see, Bodya's score is greater, so he wins the game. It can be shown that Bodya always can win this game.

,

E. Cells Arrangement

Input file: standard input
Output file: standard output
Time limit: 2 seconds
Memory limit: 256 megabytes

You are given an integer n. You choose n cells $(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)$ in the grid $n \times n$ where $1 \leq x_i \leq n$ and $1 \leq y_i \leq n$.

Let $\mathcal{H}$ be the set of **distinct** Manhattan distances between any pair of cells. Your task is to maximize the size of $\mathcal{H}$. Examples of sets and their construction are given in the notes.

If there exists more than one solution, you are allowed to output any.

Manhattan distance between cells (x_1, y_1) and (x_2, y_2) equals $|x_1 - x_2| + |y_1 - y_2|$.

Input

The first line contains a single integer t ($1 \leq t \leq 50$) — the number of test cases.

Each of the following t lines contains a single integer n ($2 \leq n \leq 10^3$).

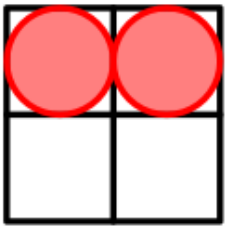
Output

For each test case, output n points which maximize the size of $\mathcal{H}$. It is not necessary to output an empty line at the end of the answer for each test case.

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

Note

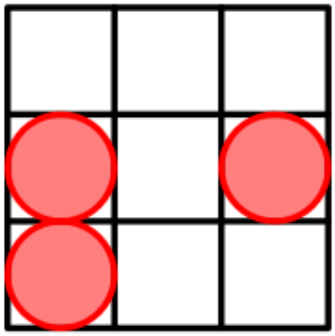
In the first testcase we have $n = 2$. One of the possible arrangements is:



The arrangement with cells located in $(1, 1)$ and $(1, 2)$.

In this case $\mathcal{H} = \{|1 - 1| + |1 - 1|, |1 - 1| + |2 - 2|, |1 - 1| + |1 - 2|\} = \{0, 0, 1\} = \{0, 1\}$. Hence, the size of $\mathcal{H}$ is 2. It can be shown that it is the greatest possible answer.

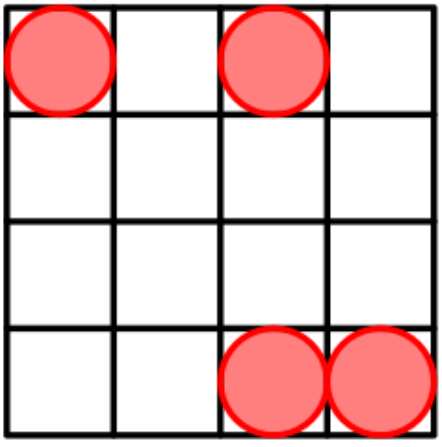
In the second testcase we have $n = 3$. The optimal arrangement is:



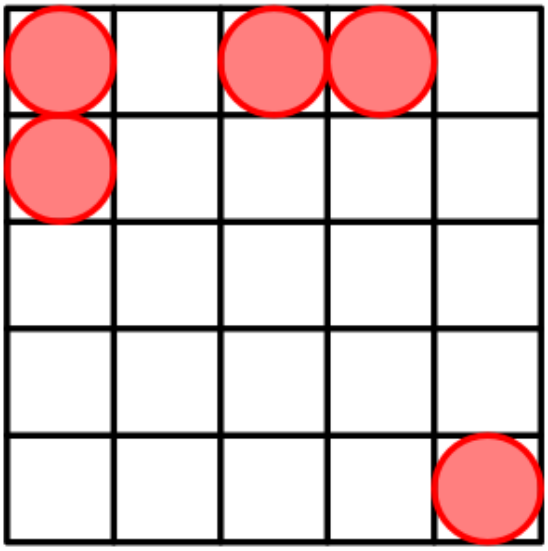
The arrangement with cells located in $(2, 1)$, $(2, 3)$ and $(3, 1)$.

$\mathcal{H}$=
$\{|2 - 2| + |1 - 1|, |2 - 2| + |3 - 3|, |3 - 3| + |1 - 1|, |2 - 2| + |1 - 3|, |2 - 3| + |1 - 1|, |2 - 3| + |3 - 1|\}$
=$\{0, 0, 0, 2, 1, 3\}$=$\{0, 1, 2, 3\}$.

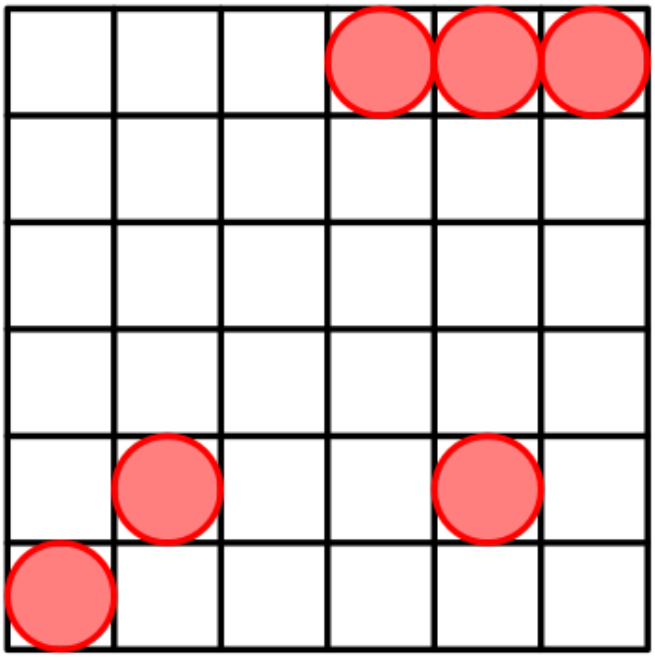
For $n = 4$ a possible arrangement is:



For $n = 5$ a possible arrangement is:



For $n = 6$ a possible arrangement is:



,

F. Equal XOR Segments

Input file: standard input
Output file: standard output
Time limit: 5 seconds
Memory limit: 256 megabytes

Let us call an array $x_1, \ldots, x_m$ interesting if it is possible to divide the array into $k > 1$ parts so that bitwise XOR of values from each part are equal.

More formally, you must split array x into k consecutive segments, each element of x must belong to **exactly** 1 segment. Let $y_1, \ldots, y_k$ be the XOR of elements from each part respectively. Then $y_1 = y_2 = \cdots = y_k$ must be fulfilled.

For example, if $x = [1, 1, 2, 3, 0]$, you can split it as follows: $[1], [1], [2, 3, 0]$. Indeed $1 = 1 = 2 \oplus 3 \oplus 0$.

You are given an array $a_1, \ldots, a_n$. Your task is to answer q queries:

- For fixed l, r, determine whether the subarray $a_l, a_{l+1}, \ldots, a_r$ is interesting.

Input

The first line contains a single integer t ($1 \le t \le 10^4$) — the number of test cases.

The first line of each test case contains two integers n and q ($2 \le n \le 2 \cdot 10^5$, $1 \le q \le 2 \cdot 10^5$) — the number of elements in the array and the number of queries respectively.

The next line contains n integers $a_1, \ldots, a_n$ ($0 \le a_i < 2^{30}$) — elements of the array.

Each of the next q lines contains two integers l and r ($1 \le l < r \le n$) describing the query.

It is guaranteed that the sum of n over all testcases does not exceed $2 \cdot 10^5$.

It is guaranteed that the sum of q over all testcases does not exceed $2 \cdot 10^5$.

Output

For each query, output "YES" if the subarray is interesting and "NO" otherwise.

You can output "Yes" and "No" in any case (for example, the strings "yES", "yes", and "Yes" will be recognized as correct answers).

Standard Input	Standard Output
4 5 5 1 1 2 3 0 1 5 2 4 3 5 1 3 3 4 5 5 1 2 3 4 5 1 5 2 4 3 5 1 3 2 3	YES YES NO NO NO YES NO NO YES NO NO NO NO

7 4 12 9 10 9 10 11 9 1 5 1 7 2 6 2 7 11 4 0 0 1 0 0 1 0 1 1 0 1 1 2 2 5 6 9 7 11	NO YES NO YES YES

Note

Explanation for the first test case:

The first query is described in the statement.

In the second query, we should divide $[1, 2, 3]$. A possible division is $[1, 2], [3]$, since $1 \oplus 2 = 3$.

It can be shown that for queries $3, 4, 5$, the subarrays are not interesting.

,

G1. Division + LCP (easy version)

Input file: standard input
Output file: standard output
Time limit: 2 seconds
Memory limit: 256 megabytes

This is the easy version of the problem. In this version $l = r$.

You are given a string s. For a fixed k, consider a division of s into exactly k continuous substrings $w_1, \ldots, w_k$. Let f_k be the maximal possible $LCP(w_1, \ldots, w_k)$ among all divisions.

$LCP(w_1, \ldots, w_m)$ is the length of the Longest Common Prefix of the strings $w_1, \ldots, w_m$.

For example, if $s = abababcab$ and $k = 4$, a possible division is $ab|ab|abc|ab$. The $LCP(ab, ab, abc, ab)$ is 2, since ab is the Longest Common Prefix of those four strings. Note that each substring consists of a continuous segment of characters and each character belongs to **exactly** one substring.

Your task is to find $f_l, f_{l+1}, \ldots, f_r$. **In this version** $l = r$.

Input

The first line contains a single integer t ($1 \le t \le 10^4$) — the number of test cases.

The first line of each test case contains two integers n, l, r ($1 \le l = r \le n \le 2 \cdot 10^5$) — the length of the string and the given range.

The second line of each test case contains string s of length n, all characters are lowercase English letters.

It is guaranteed that the sum of n over all test cases does not exceed $2 \cdot 10^5$.

Output

For each test case, output $r - l + 1$ values: $f_l, \ldots, f_r$.

Standard Input	Standard Output
7 3 3 3 aba 3 3 3 aaa 7 2 2 abacaba 9 4 4 abababcab 10 1 1 codeforces 9 3 3 abafababa 5 3 3 zpozp	0 1 3 2 10 2 0

Note

In the first sample $n = k$, so the only division of aba is $a|b|a$. The answer is zero, because those strings do not have a common prefix.

In the second sample, the only division is $a|a|a$. Their longest common prefix is one.

,

G2. Division + LCP (hard version)

Input file: standard input
Output file: standard output
Time limit: 3 seconds
Memory limit: 256 megabytes

This is the hard version of the problem. In this version $l \le r$.

You are given a string s. For a fixed k, consider a division of s into exactly k continuous substrings $w_1, \ldots, w_k$. Let f_k be the maximal possible $LCP(w_1, \ldots, w_k)$ among all divisions.

$LCP(w_1, \ldots, w_m)$ is the length of the Longest Common Prefix of the strings $w_1, \ldots, w_m$.

For example, if $s = abababcab$ and $k = 4$, a possible division is $ab|ab|abc|ab$. The $LCP(ab, ab, abc, ab)$ is 2, since ab is the Longest Common Prefix of those four strings. Note that each substring consists of a continuous segment of characters and each character belongs to **exactly** one substring.

Your task is to find $f_l, f_{l+1}, \ldots, f_r$.

Input

The first line contains a single integer t ($1 \le t \le 10^4$) — the number of test cases.

The first line of each test case contains two integers n, l, r ($1 \le l \le r \le n \le 2 \cdot 10^5$) — the length of the string and the given range.

The second line of each test case contains string s of length n, all characters are lowercase English letters.

It is guaranteed that the sum of n over all test cases does not exceed $2 \cdot 10^5$.

Output

For each test case, output $r - l + 1$ values: $f_l, \ldots, f_r$.

Standard Input	Standard Output
7 3 1 3 aba 3 2 3 aaa 7 1 5 abacaba 9 1 6 abababcab 10 1 10 aaaaaaawac 9 1 9 abafababa 7 2 7 vvzvvvv	3 1 0 1 1 7 3 1 1 0 9 2 2 2 0 0 10 3 2 1 1 1 1 1 0 0 9 3 2 1 1 0 0 0 0 2 2 1 1 1 0