Problem A
ABC Stamp

Time limit: 1 second
Memory limit: 1.5 Gigabytes

Problem Description

There is a string S of length N. Initially, S_i is `?` for all i.

You can perform the following operation any number of times.

- Select an index i such that $1 \le i \le N-2$, and change the following three characters:
  - $S_i :=$ `A`
  - $S_{i+1} :=$ `B`
  - $S_{i+2} :=$ `C`

You are given a target string R of length N, consisting of letters `A`, `B`, `C` only.

Please determine if it is possible to make S equal to R by using the operation any number of times.

If it is found to be possible, construct a sequence of at most N operations that makes S equal to R. It can be proven that if a sequence exists, there also exists a sequence of size at most N.

Note that you do **not** need to minimize the length of the sequence.

Input Format

- The first line of input will contain a single integer T, denoting the number of test cases.
- Each test case consists of two lines of input.
  - The first line of each test case contains N - the length of the strings S and R.
  - The second line contains the string R.

Output Format

For each test case, output the following:

- If it is possible to convert S to R,
  - First, on a new line, output K $(1 \le K \le N)$ - the number of operations used.
  - Then, on another line, output a sequence of K integers, $i_1, i_2, \ldots, i_K$ $(1 \le i_j \le N-2)$ representing the parameter i for each of the K operations.
- Otherwise, output the single integer -1 on a new line.

Constraints

- $1 \le T \le 10^4$
- $3 \le N \le 2 \cdot 10^5$
- $R_i \in \{\texttt{A}, \texttt{B}, \texttt{C}\}$
- $|R| = N$
- The sum of N over all test cases does not exceed $2 \cdot 10^5$

Samples

Sample Input 1

```
3
3
ABC
5
AABCC
3
BBC
```

Sample Output 1

```
1
1
3
1 3 2
-1
```

Sample Explanation

Test case 1: Initially $S =$ `???`, and after the operation with $i = 1$ it becomes `ABC`, which equals R.

Test case 2: Initially $S =$ `?????`. The sequence of moves is as follows:

- First, $i = 1$ is chosen. Now, $S =$ `ABC??`.
- Next, $i = 3$ is chosen. Now, $S =$ `ABABC`.
- Finally, $i = 2$ is chosen. Now, $S =$ `AABCC`, which equals R.

Test case 3: No valid sequence of moves exists.

Problem B
Arithmetic Subarrrays

Time limit: 2 seconds
Memory limit: 1.5 Gigabytes

Problem Description

Let's define the array $[B_1, B_2, \ldots, B_M]$ to be *good* if there exists a permutation of $[B_1, B_2, \ldots, B_M]$ that is an arithmetic progression.

Formally, the array is *good* if there exist arrays C and D such that all of the following are satisfied:

- $|C| = |D| = |B| = M$.
- C is a permutation of the integers from 1 to M. That is, each of the numbers from 1 to M appears exactly once in the array C.
- $D_i = B_{C_i}$ for all integers $1 \le i \le M$.
- $D_{i+1} - D_i$ is constant over all integers $1 \le i < M$.

You are given an array A of length N and Q queries of the form (L, R). For each query, you must print whether the array $[A_L, A_{L+1}, A_{L+2}, \ldots, A_R]$ is *good*.

The queries will be given to you in an encrypted format. Let's suppose that in the current test case, there are X queries so far whose answer is `Yes`, and the query given to you is (L', R'). Then, the decrypted query has bounds $((L' + X) \bmod N + 1, (R' + X) \bmod N + 1)$.

Input Format

- The first line of input will contain a single integer T, denoting the number of test cases.
- Each test case consists of multiple lines of input.
  - The first line of each test case contains N and Q - the length of the array and the number of queries.
  - The second line contains N integers - $A_1, A_2, \ldots, A_N$.
  - The i-th of next Q lines each contain the encrypted parameters for the i-th query, i.e. L' and R'.

Output Format

For each test case, for each query, output on a new line if the decrypted subarray (L, R) is *good* or not.

Each character of the output may be printed in either uppercase or lowercase, i.e., the strings `no`, `No`, `nO`, and `NO` will all be treated as identical.

Constraints

- $1 \le T \le 10^4$
- $1 \le N, Q \le 2 \cdot 10^5$
- $1 \le A_i \le 10^9$
- $0 \le L', R' < N$

- The decrypted parameters L and R follow $1 \leq L < R \leq N$
- The sum of N and the sum of Q does not exceed $2 \cdot 10^5$

Samples

Sample Input 1

```
2
5 4
2 1 3 5 4
1 4
0 1
4 1
3 2
3 3
1 1 2
0 2
0 1
0 1
```

Sample Output 1

```
No
Yes
Yes
Yes
No
Yes
Yes
```

Sample Explanation

Test case 1: The decrypted queries are as follows:

- $L = 2, R = 5$
- $L = 1, R = 2$
- $L = 1, R = 3$
- $L = 1, R = 5$

It can be verified that all except the first are *good*.

Test case 2: The decrypted queries are as follows:

- $L = 1, R = 3$
- $L = 1, R = 2$
- $L = 2, R = 3$

Problem C
Balanced Subsequence

Time limit: 1 second
Memory limit: 1.5 Gigabytes

Problem Description

We define a set of **balanced** strings in the following way:

- () and the empty string are balanced.
- If A is balanced, so is (A).
- If A and B are balanced, so is $A + B$.
  - $A + B$ denotes the concatenation of A and B.

Given a string S, let $f(S)$ = the length of the longest balanced string, which is a subsequence of S. Note that a subsequence need not be continuous.

You are given a string S of length N consisting of (and) only. Find the N values $f(S[1,1]), f(S[1,2]), f(S[1,3]), \ldots, f(S[1,N])$, i.e., solve the problem for every prefix.

Input Format

- The first line of input will contain a single integer T, denoting the number of test cases.
- Each test case consists of two lines of input.
  - The first line contains N - the length of the string.
  - The second line contains S - the string.

Output Format

For each test case, output on a new line N answers, $f(S[1,1]), f(S[1,2]), f(S[1,3]), \ldots, f(S[1,N])$.

Constraints

- $1 \leq T \leq 10^4$
- $1 \leq N \leq 2 \cdot 10^5$
- $|S| = N$
- - $S_i ='('$ or $S_i =')'$
- The sum of N over all test cases does not exceed $2 \cdot 10^5$.

Samples

Sample Input 1

```
3
5
())()
4
)(()
6
(())()
```

Sample Output 1

```
0 2 2 2 4
0 0 0 2
0 0 2 4 4 6
```

Sample Explanation

Test case 1: Here are the answers for each prefix:

- (: There is no valid balanced subsequence.
- () : We can choose the whole sequence as a valid balanced subsequence, so the answer is 2.
- ()) : We choose the 1^{st} and 2^{nd} characters to get ().
- ())(: We choose the 1^{st} and 2^{nd} characters to get ().
- ())() : We choose the $1^{st}, 2^{nd}, 4^{th}$ and 5^{th} characters to get ()(), which is length 4 and balanced.

Problem D
Bin Packing

Time limit: 10 seconds
Memory limit: 1.5 Gigabytes

Problem Description

Let's solve an NP Hard problem today.

There are N objects, the i-th of which has size A_i ($1 \leq A_i \leq 12$). You have several bins of size 12. Multiple objects can be fit into the same bin if and only if the sum of their sizes is at most 12.

Find the minimum number of bins needed to fit all N objects.

There is a **special constraint** on the input.

1. For all non-sample files, the number of test cases T and the number of objects N are fixed. ($T = 100, N = 1000$).
2. The object sizes are generated randomly. Formally, each A_i is independently chosen from a uniform distribution on the set of integers in the interval $[1, 12]$.

Please note the really large TL and so it may take a while to judge. We recommend you to run your code locally on a max test to estimate it's running time beforehand.

Input Format

- The first line of input will contain a single integer T, denoting the number of test cases.
- Each test case consists of two lines of input.
  - The first line of each test case contains N - the number of objects.
  - The second line contains N integers - $A_1, A_2, \ldots, A_N$ - the sizes of the objects.

Output Format

For each test case, output on a new line the minimum number of bins needed.

Constraints

For all non-sample tests:

- $T = 100$
- $N = 1000$
- $1 \leq A_i \leq 12$
- Each A_i is a uniformly randomly generated integer in the range $[1, 12]$.

There are exactly 19 non-sample tests.

Samples

Sample Input 1

```
10
5
8 2 2 3 9
3
11 1 1
8
9 8 2 3 4 2 2 12
10
2 2 2 2 2 2 3 3 3 3
6
5 2 8 4 1 11
8
12 12 9 8 2 8 4 4
5
5 5 6 6 6
8
10 12 5 4 8 2 2 11
7
5 1 5 7 9 4 4
6
12 1 1 9 3 5
```

Sample Output 1

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

Sample Explanation

Test case 1: We can distribute the objects like this:

- Bin 1: Objects 1, 2 and 3. The sum of their sizes is exactly 12.
- Bin 2: Objects 4 and 5. The sum of their sizes is also exactly 12.

Thus, 2 bins suffice.

Problem E
Cascading Operations

Time limit: 1 second
Memory limit: 1.5 Gigabytes

Problem Description

You are given a string S of length N consisting of lowercase English characters only. It is guaranteed that $S_i \neq S_{i+1}$ for all $1 \leq i < N$.

You must insert a single lowercase character somewhere into the string. Afterwards, the following process happens:

- While there exists an index i ($1 \leq i < |S|$) such that $S_i = S_{i+1}$, such an index i is found and we delete both S_i and S_{i+1} from the string.
  - Note that it can be shown that at any moment there will not exist multiple such indices.

Please determine if it is possible to choose the character and location appropriately so that the string becomes **empty** at the end of the process.

Input Format

- The first line of input will contain a single integer T, denoting the number of test cases.
- Each test case consists of two lines of input.
  - The first line of each test case contains N - the length of the string.
  - The second line contains S - the string.

Output Format

For each test case, output on a new line `Yes` if it is possible, and `No` otherwise.

Each character can be outputted in either case, i.e., `yes`, `YES`, `yEs` will all be accepted as positive responses.

Constraints

- $1 \leq T \leq 10^4$
- $1 \leq N \leq 2 \cdot 10^5$
- $S_i \neq S_{i+1}$
- $|S| = N$
- S contains lowercase English characters only.
- The sum of N over all test cases does not exceed $2 \cdot 10^5$.

Samples

Sample Input 1

```
2
3
aba
3
abc
```

Sample Output 1

```
Yes
No
```

Sample Explanation

Test case 1: Insert `b` into the string at the second position, to obtain `abba`. Then:

- First, since $S_2 = S_3 = \texttt{b}$, these two characters are deleted from S. S is now `aa`.
- Next, since $S_1 = S_2 = \texttt{a}$, these two characters are deleted from S and it becomes empty.

Problem F
Easy Counting Problem

Time limit: 2 seconds
Memory limit: 1.5 Gigabytes

Problem Description

Did I just copy a problem from Kanpur regionals and ask you to solve it again? Yes, I did. Sue me.

Alice had an array A of N elements, but for no reason whatsoever, a monster has started attacking the array. (Why? How? We don't know either!)

The monster does not like fixed indices (where a fixed index means that $A_i = i$). Thus, every second starting from 1, the monster deletes the **maximum** index i such that $A_i = i$ (if such an index exists).

Please note that in this problem, the monster is only deleting 1 element per attack. Also, the monster is deleting fixed indices instead.

The remaining elements are then re-indexed starting from 1 without changing their order. Then, the monster will again attack next second if there are more elements it can delete, and so on.

Alice wants to know when the monster will stop attacking her array. The monster will stop attacking when there are no more elements it can delete. The state of the array is called stable then. Note that an empty array is said to be stable as well.

Let $f(A)$ denote the time taken for the array to become stable. If the array was stable from the beginning, $f(A) = 0$.

Given integers N and M, find the sum of $f(A)$ over all M^N integer arrays satisfying $|A| = N$ and $1 \leq A_i \leq M$ (Array length is N and all elements are integers between 1 and M). Since the answer may be large, output it modulo 998244353.

Input Format

- The first line of input will contain a single integer T, denoting the number of test cases.
- The first and only line of input for each test case contains 2 integers N and M.

Output Format

For each test case, output on a new line the sum of $f(A)$ over all arrays of length N and elements bounded by M, modulo 998244353.

Constraints

- $1 \leq T \leq 10^4$
- $1 \leq N, M \leq 2 \cdot 10^5$
- $|N - M| \leq 20$
- The sum of N and the sum of M both do not exceed $2 \cdot 10^5$.

Samples

Sample Input 1

```
9
2 1
2 2
3 1
3 2
3 3
3 5
6 3
23 10
190000 190000
```

Sample Output 1

```
2
5
3
15
37
91
1840
392207909
646824155
```

Sample Explanation

Test case 2: There are $2^2 = 4$ possible arrays. Their respective values are:

- $f([1,1]) = 2$
  - On the first move, the element at index 1 will be deleted and the array becomes [1].
  - On the second move, the element at index 1 will again be deleted and the array becomes empty, and hence stable.
- $f([1,2]) = 2$
  - On the first move, the element at index 2 will be deleted and the array becomes [1].
  - On the second move, the element at index 1 will be deleted and the array becomes empty, and hence stable.
- $f([2,1]) = 0$
  - The array is already stable.
- $f([2,2]) = 1$
  - On the first move, the element at index 2 will be deleted and the array becomes [2], which is stable.

The sum of these values is $2 + 2 + 0 + 1 = 5$.

Problem G
Good Sequence

Time limit: 1 second
Memory limit: 1.5 Gigabytes

Problem Description

A sequence B of length M is said to be *good* if the number of indices i such that $B_i = i$ is at most $\frac{M}{2}$.

Given a sequence A of length N, you want to find the length of the longest *good* subsequence of A. Please find this value.
If no non-empty good subsequence exists, print 0.

Input Format

- The first line of input will contain a single integer T, denoting the number of test cases.
- Each test case consists of two lines of input.
  - The first line contains N - the length of the sequence.
  - The second line contains N integers - $A_1, A_2, \ldots, A_N$.

Output Format

For each test case, output on a new line the length of the longest *good* subsequence.

Constraints

- $1 \leq T \leq 10^4$
- $1 \leq N \leq 2 \cdot 10^5$
- $1 \leq A_i \leq N$
- The sum of N over all test cases does not exceed $2 \cdot 10^5$.

Samples

Sample Input 1

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

Sample Output 1

```
0
2
```

Sample Explanation

Test case 1: $[1]$ is not a good subsequence because one element equals its index, which is more than $\frac{1}{2}$.
No non-empty good subsequence exists, so the answer is 0.

Test case 2: A is already good, so the answer is 2.

Problem H
How Odd

Time limit: 1 second
Memory limit: 1.5 Gigabytes

Problem Description

You are given a simple undirected graph G with N vertices and M edges.
It is further guaranteed that **every simple cycle**† **in** G **has even length**.

Let H denote the graph that is the *complement* of G — that is, H is a simple undirected graph on N vertices which contains the edge (u, v) if and only if G does not contain (u, v).

Find any simple cycle in H whose length is odd, or claim that none exist.

---

† A simple cycle of length k ($k \geq 3$) is a sequence of vertices $(v_1, v_2, \ldots, v_k)$ such that:

- $v_i \neq v_j$ for any pair (i, j) satisfying $1 \leq i < j \leq k$,
- There exists an edge between v_i and v_{i+1} for each $1 \leq i < k$, and
- There exists an edge between v_1 and v_k.

Input Format

- The first line of input will contain a single integer T, denoting the number of test cases.
- Each test case consists of multiple lines of input.
  - The first line of each test case contains two space-separated integers N and M — the number of vertices and edges of the graph, respectively.
  - The next M lines describe the edges. The i-th of these M lines contains two space-separated integers u_i and v_i, denoting an edge between u_i and v_i.

Output Format

For each test case:

- If no odd simple cycle exists in H, print the single integer -1 on a new line.
- Otherwise, print two lines:
  - On the first line, print an integer K ($K \geq 3$) denoting the size of the odd simple cycle you found.
  - On the second line, print K space-separated integers $v_1, v_2, \ldots, v_K$ denoting the vertices of the cycle, in order. There must not exist an edge in G between v_i and $v_{i \bmod K+1}$ for each $1 \leq i \leq K$.

If there are multiple possible answers, you may find any of them.

Constraints

- $1 \leq T \leq 10^5$
- $1 \leq N \leq 2 \cdot 10^5$
- $0 \leq M \leq \min\left(\frac{N \cdot (N-1)}{2}, 2 \cdot 10^5\right)$
- $1 \leq u_i, v_i \leq N$
- $u_i \neq v_i$ for each $1 \leq i \leq M$.
- All input edges are distinct.
- The sum of N and the sum of M across all test cases both won't exceed $2 \cdot 10^5$.

Samples

Sample Input 1

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

Sample Output 1

```
-1
-1
3
3 4 1
```

Sample Explanation

Test case 1: The complement graph of G contains no edges. It can be verified that no odd cycle exists.

Test case 3: The complement graph of G contains the edges $\{(1,3),(1,4),(3,4)\}$.
$(3,4,1)$ is a cycle of length 3 (which is odd) in such a graph.

Problem I
MEX is Fun

Time limit: 2 seconds
Memory limit: 1.5 Gigabytes

Problem Description

Given two sequences A and B of length N, you need to perform the following operation to construct an array C of length N:

- For all indices i $(1 \le i \le N)$, set C_i to either A_i or B_i.

Your task is to find the maximum value of $\texttt{MEX}(C_1, C_2, \ldots, C_i)$ for each $1 \le i \le N$ **independently**.

Here, `MEX` (Minimum Excludant) of a sequence is the smallest non-negative integer that is not present in the sequence.

Input Format

- The first line of input will contain a single integer T, denoting the number of test cases.
- Each test case consists of three lines of input.
  - The first line of each test case contains an integer N.
  - The second line of each test case contains N space-separated integers $A_1, A_2, \ldots, A_N$.
  - The third line of each test case contains N space-separated integers $B_1, B_2, \ldots, B_N$.

Output Format

For each test case, output a single line containing N integers. The i-th integer in the output corresponds to the maximum value of $\texttt{MEX}(C_1, C_2, \ldots, C_i)$.

Constraints

- $1 \le T \le 10^5$
- $2 \le N \le 4 \cdot 10^5$
- $0 \le A_i, B_i \le N$
- The sum of N over all test cases won't exceed $4 \cdot 10^5$.

Samples

Sample Input 1

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

Sample Output 1

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

Sample Explanation

Test case 1:

- For $i = 1$, one possible array is $C = [1]$. The `MEX` value of C is 0;
- For $i = 2$, one possible array is $C = [1, 2]$. The `MEX` value of C is 0;
- For $i = 3$, one possible array is $C = [1, 2, 0]$. The `MEX` value of C is 3;
- For $i = 4$, one possible array is $C = [1, 2, 3, 0]$. The `MEX` value of C is 4.

Problem J
Recursion

Time limit: 1 second
Memory limit: 1.5 Gigabytes

Problem Description

Recursively define the sequence $X_1 = 1$, $X_{2 \cdot i} = X_i + 1$, and $X_{2 \cdot i+1} = \frac{1}{X_{2 \cdot i}}$.

Next, define Y_i as the smallest positive integer such that $\frac{Y_i}{X_i}$ is an integer. It can be proven that such an integer always exists.

You are given 2 integers L and R. Find $\sum_{i=L}^{R} Y_i$. Since the answer might be too large, output it modulo 998244353.

Input Format

- The first line of input will contain a single integer T, denoting the number of test cases.
- The first and only line of each test case contains 2 integers - L and R.

Output Format

For each test case, output on a new line $\sum_{i=L}^{R} Y_i$ modulo 998244353.

Constraints

- $1 \le T \le 10^4$
- $1 \le L \le R \le 10^{18}$

Samples

Sample Input 1

```
4
3 6
2 8
1 4
14 34
```

Sample Output 1

```
8
16
7
102
```

Sample Explanation

The first few terms of the sequence X are as follows:

- $X_1 = 1$
- $X_2 = X_1 + 1 = 2$
- $X_3 = \frac{1}{X_2} = \frac{1}{2}$
- $X_4 = X_2 + 1 = 3$
- $X_5 = \frac{1}{X_4} = \frac{1}{3}$
- $X_6 = X_3 + 1 = \frac{3}{2}$

Consequently, the first few terms of the sequence Y are as follows:

- $Y_1 = 1$
- $Y_2 = 2$
- $Y_3 = 1$
- $Y_4 = 3$
- $Y_5 = 1$
- $Y_6 = 3$

Problem K
The Perfect Tour

Time limit: 2 seconds
Memory limit: 1.5 Gigabytes

Problem Description

There is a grid of $N \cdot M$ cells. It is guaranteed that $N \cdot M$ is even. Exactly half of the cells, i.e., $\frac{N \cdot M}{2}$ are coloured white. The rest are coloured black.

Construct 2 directed tournament[†] graphs: one on the black cells, and another on the white cells. There are a total of $C\left(\frac{N \cdot M}{2}, 2\right)^{\ddagger}$ edges in each tournament graph. Thus, a total of $2 \cdot C\left(\frac{N \cdot M}{2}, 2\right)$ edges.

Your task is to direct these edges to ensure that the resultant vector sum of all these edges is $\vec{0}$. If an edge is directed from cell (a, b) to (c, d), the corresponding vector is $(c - a)\hat{i} + (d - b)\hat{j}$.

[†]A directed tournament of size k is a graph on k vertices such that for each $1 \leq i < j \leq k$, exactly one of the edges (i, j) and (j, i) exists in the graph.

[‡]$C(x, y)$ denotes the binomial coefficient $\binom{x}{y}$.

Input Format

- The first line of input contains N and M - the dimensions of the grid.
- The next N lines contain M characters each - the colour of each cell in the grid. 0 represents white, and 1 represents black.

Output Format

For each test case:

- If the answer is possible, output 1, otherwise 0.
- If you answered 1, also output $2 \cdot C\left(\frac{N \cdot M}{2}, 2\right)$ lines. Each of these lines should have 4 integers r_1, c_1, r_2, c_2 representing a directed edge from (r_1, c_1) to (r_2, c_2).

The 2 outputted cells on each line should be distinct cells, and have the same colour. Further, no pair of cells should be repeated. You can output the pairs in any order you want.

Constraints

- $1 \leq N, M \leq 1000$
- $2 \leq N \cdot M \leq 1000$
- $N \cdot M$ is even
- All grid colours are 0 or 1
- There are exactly $\frac{N \cdot M}{2}$ 0s.

Samples

Sample Input 1

```
1 6
010101
```

Sample Output 1

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

Sample Input 2

```
2 2
01
01
```

Sample Output 2

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

Sample Input 3

```
2 6
101010
010101
```

Sample Output 3

```
0
```

Problem L
XOR 123

Time limit: 4 seconds
Memory limit: 1.5 Gigabytes

Problem Description

There is a sequence A of N integers. Initially, all $A_i = 0$.

There will be a total of Q updates to the sequence, of the following form:

- Given L and R, for all $L \le i \le R$, set A_i to $A_i \oplus (i - L + 1)$, where $\oplus$ represents the Bitwise XOR operator.

Find the final sequence A after all Q updates.

Input Format

- The first line of input will contain a single integer T, denoting the number of test cases.
- Each test case consists of multiple lines of input.
  - The first line contains 2 integers - N and Q, the size of the array and the number of updates.
  - The next Q lines each contain 2 integers - L and R, parameters of the update.

Output Format

For each test case, output on a new line N integers - the final array A after all updates.

Constraints

- $1 \le T \le 10^4$
- $1 \le N, Q \le 5 \cdot 10^5$
- $1 \le L \le R \le N$
- The sum of N and the sum of Q both do not exceed $5 \cdot 10^5$.

Samples

Sample Input 1

```
3
4 2
1 4
2 3
5 3
1 4
3 5
3 3
10 5
1 4
2 3
6 10
5 8
7 9
```

Sample Output 1

```
1 3 1 4
1 2 3 6 3
1 3 1 4 1 3 0 5 7 5
```

Sample Explanation

Test case 1: The process is as follows:

- Initially, $A = [0, 0, 0, 0]$.
- The first update has $L = 1$ and $R = 4$. After this update, $A = [1, 2, 3, 4]$.
- The second update has $L = 2$ and $R = 3$. After this update, $A = [1, 3, 1, 4]$.