A. The Third Three Number Problem

time limit per test: 1 second
memory limit per test: 256 megabytes
input: standard input
output: standard output

You are given a positive integer n. Your task is to find **any** three integers a, b and c ($0 \le a, b, c \le 10^9$) for which $(a \oplus b) + (b \oplus c) + (a \oplus c) = n$, or determine that there are no such integers.

Here $a \oplus b$ denotes the bitwise XOR of a and b. For example, $2 \oplus 4 = 6$ and $3 \oplus 1 = 2$.

Input

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

The only line of each test case contains a single integer n ($1 \le n \le 10^9$).

Output

For each test case, print **any** three integers a, b and c ($0 \le a, b, c \le 10^9$) for which $(a \oplus b) + (b \oplus c) + (a \oplus c) = n$. If no such integers exist, print -1.

Example

input

```
5
4
1
12
2046
194723326
```

output

```
3 3 1
-1
2 4 6
69 420 666
12345678 87654321 100000000
```

Note

In the first test case, $a = 3$, $b = 3$, $c = 1$, so $(3 \oplus 3) + (3 \oplus 1) + (3 \oplus 1) = 0 + 2 + 2 = 4$.

In the second test case, there are no solutions.

In the third test case, $(2 \oplus 4) + (4 \oplus 6) + (2 \oplus 6) = 6 + 2 + 4 = 12$.

B. Almost Ternary Matrix

time limit per test: 1 second
memory limit per test: 256 megabytes
input: standard input
output: standard output

You are given two **even** integers n and m. Your task is to find **any** binary matrix a with n rows and m columns where every cell (i, j) has **exactly** two neighbours with a different value than $a_{i,j}$.

Two cells in the matrix are considered neighbours if and only if they share a side. More formally, the neighbours of cell (x, y) are: $(x-1, y)$, $(x, y+1)$, $(x+1, y)$ and $(x, y-1)$.

It can be proven that under the given constraints, an answer always exists.

Input

Each test contains multiple test cases. The first line of input contains a single integer t ($1 \le t \le 100$) — the number of test cases. The following lines contain the descriptions of the test cases.

The only line of each test case contains two **even** integers n and m ($2 \le n, m \le 50$) — the height and width of the binary matrix, respectively.

Output

For each test case, print n lines, each of which contains m numbers, equal to 0 or 1 — any binary matrix which satisfies the constraints described in the statement.

It can be proven that under the given constraints, an answer always exists.

Example

input

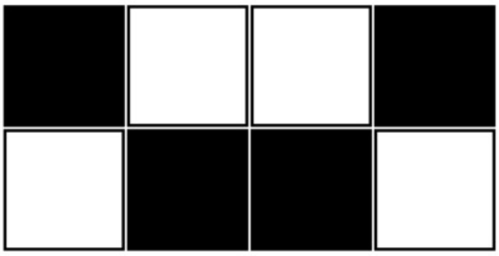
```
3
2 4
2 2
4 4
```

output

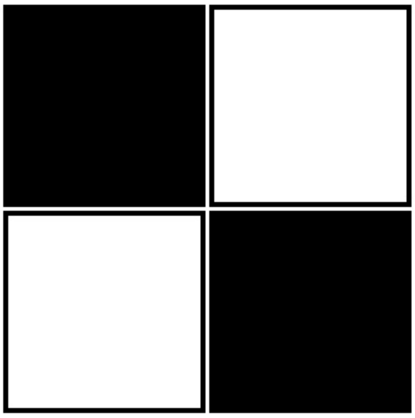
```
1 0 0 1
0 1 1 0
1 0
0 1
1 0 1 0
0 0 1 1
1 1 0 0
0 1 0 1
```

Note

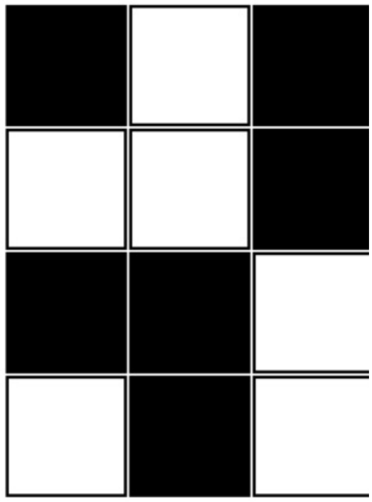
White means 0, black means 1.



The binary matrix from the first test case



The binary matrix from the second test case



The binary matrix from the third test case

C. The Third Problem

time limit per test: 1 second
memory limit per test: 256 megabytes
input: standard input
output: standard output

You are given a permutation $a_1, a_2, \ldots, a_n$ of integers from 0 to $n-1$. Your task is to find how many permutations $b_1, b_2, \ldots, b_n$ are *similar* to permutation a.

Two permutations a and b of size n are considered *similar* if for all intervals $[l, r]$ ($1 \le l \le r \le n$), the following condition is satisfied:

$$\operatorname{MEX}([a_l, a_{l+1}, \ldots, a_r]) = \operatorname{MEX}([b_l, b_{l+1}, \ldots, b_r]),$$

where the MEX of a collection of integers $c_1, c_2, \ldots, c_k$ is defined as the smallest non-negative integer x which does not occur in collection c. For example, $\operatorname{MEX}([1, 2, 3, 4, 5]) = 0$, and $\operatorname{MEX}([0, 1, 2, 4, 5]) = 3$.

Since the total number of such permutations can be very large, you will have to print its remainder modulo $10^9 + 7$.

In this problem, a permutation of size n is an array consisting of n distinct integers from 0 to $n-1$ in arbitrary order. For example, $[1, 0, 2, 4, 3]$ is a permutation, while $[0, 1, 1]$ is not, since 1 appears twice in the array. $[0, 1, 3]$ is also not a permutation, since $n = 3$ and there is a 3 in the array.

Input

Each test contains multiple test cases. The first line of input contains one integer t ($1 \le t \le 10^4$) — the number of test cases. The following lines contain the descriptions of the test cases.

The first line of each test case contains a single integer n ($1 \le n \le 10^5$) — the size of permutation a.

The second line of each test case contains n distinct integers $a_1, a_2, \ldots, a_n$ ($0 \le a_i < n$) — the elements of permutation a.

It is guaranteed that the sum of n across all test cases does not exceed 10^5.

Output

For each test case, print a single integer, the number of permutations *similar* to permutation a, taken modulo $10^9 + 7$.

Example

input

```
5
5
4 0 3 2 1
1
0
4
0 1 2 3
6
1 2 4 0 5 3
8
1 3 7 2 5 0 6 4
```

output

```
2
1
1
4
72
```

Note

For the first test case, the only permutations similar to $a = [4, 0, 3, 2, 1]$ are $[4, 0, 3, 2, 1]$ and $[4, 0, 2, 3, 1]$.

For the second and third test cases, the given permutations are only similar to themselves.

For the fourth test case, there are 4 permutations similar to $a = [1, 2, 4, 0, 5, 3]$:

- $[1, 2, 4, 0, 5, 3]$;
- $[1, 2, 5, 0, 4, 3]$;

- $[1, 4, 2, 0, 5, 3]$;
- $[1, 5, 2, 0, 4, 3]$.

D. Almost Triple Deletions

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

You are given an integer n and an array $a_1, a_2, \ldots, a_n$.

In one operation, you can choose an index i ($1 \le i < n$) for which $a_i \ne a_{i+1}$ and delete both a_i and a_{i+1} from the array. After deleting a_i and a_{i+1}, the remaining parts of the array are concatenated.

For example, if $a = [1, 4, 3, 3, 6, 2]$, then after performing an operation with $i = 2$, the resulting array will be $[1, 3, 6, 2]$.

What is the maximum possible length of an array of **equal** elements obtainable from a by performing several (perhaps none) of the aforementioned operations?

Input

Each test contains multiple test cases. The first line of input contains one integer t ($1 \le t \le 1000$) — the number of test cases. The following lines contain the descriptions of the test cases.

The first line of each test case contains a single integer n ($1 \le n \le 5000$) — the length of array a.

The second line of each test case contains n integers $a_1, a_2, \ldots, a_n$ ($1 \le a_i \le n$) — the elements of array a.

It is guaranteed that the sum of n across all test cases does not exceed $10\,000$.

Output

For each testcase, print a single integer, the maximum possible length of an array of **equal** elements obtainable from a by performing a sequence of operations.

Example

input

```
5
7
1 2 3 2 1 3 3
1
1
6
1 1 1 2 2 2
8
1 1 2 2 3 3 1 1
12
1 5 2 3 3 3 4 4 4 4 3 3
```

output

```
3
1
0
4
2
```

Note

For the first testcase, an optimal sequence of operations would be: $[1, 2, 3, 2, 1, 3, 3] \to [3, 2, 1, 3, 3] \to [3, 3, 3]$.

For the second testcase, all elements in the array are already equal.

For the third testcase, the only possible sequence of operations is: $[1, 1, 1, 2, 2, 2] \to [1, 1, 2, 2] \to [1, 2] \to []$. Note that, according to the statement, the elements deleted at each step must be different.

For the fourth testcase, the optimal sequence of operations is: $[1, 1, 2, 2, 3, 3, 1, 1] \to [1, 1, 2, 3, 1, 1] \to [1, 1, 1, 1]$.

For the fifth testcase, one possible reachable array of two equal elements is $[4, 4]$.

E. Three Days Grace

time limit per test: 4 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

Ibti was thinking about a good title for this problem that would fit the round theme (numerus ternarium). He immediately thought about the third derivative, but that was pretty lame so he decided to include the best band in the world — Three Days Grace.

You are given a multiset A with initial size n, whose elements are integers between 1 and m. In one operation, do the following:

- select a value x from the multiset A, then
- select two integers p and q such that $p, q > 1$ and $p \cdot q = x$. Insert p and q to A, delete x from A.

Note that the size of the multiset A increases by 1 after each operation.

We define the balance of the multiset A as $\max(a_i) - \min(a_i)$. Find the minimum possible balance after performing any number (possible zero) of operations.

Input

The first line of the input contains a single integer t ($1 \le t \le 10^5$) — the number of test cases.

The second line of each test case contains two integers n and m ($1 \le n \le 10^6$, $1 \le m \le 5 \cdot 10^6$) — the initial size of the multiset, and the maximum value of an element.

The third line of each test case contains n integers $a_1, a_2, \ldots, a_n$ ($1 \le a_i \le m$) — the elements in the initial multiset.

It is guaranteed that the sum of n across all test cases does not exceed 10^6 and the sum of m across all test cases does not exceed $5 \cdot 10^6$.

Output

For each test case, print a single integer — the minimum possible balance.

Example

input

```
4
5 10
2 4 2 4 2
3 50
12 2 3
2 40
6 35
2 5
1 5
```

output

```
0
1
2
4
```

Note

In the first test case, we can apply the operation on each of the 4s with $(p,q) = (2,2)$ and make the multiset $\{2,2,2,2,2,2,2\}$ with balance $\max(\{2,2,2,2,2,2,2\}) - \min(\{2,2,2,2,2,2,2\}) = 0$. It is obvious we cannot make this balance less than 0.

In the second test case, we can apply an operation on 12 with $(p,q) = (3,4)$. After this our multiset will be $\{3,4,2,3\}$. We can make one more operation on 4 with $(p,q) = (2,2)$, making the multiset $\{3,2,2,2,3\}$ with balance equal to 1.

In the third test case, we can apply an operation on 35 with $(p,q) = (5,7)$. The final multiset is $\{6,5,7\}$ and has a balance equal to $7-5=2$.

In the forth test case, we cannot apply any operation, so the balance is $5-1=4$.