Problem A
Construct uwu

Time limit: 4 seconds
Memory limit: 1.5 Gigabytes

Problem Description

You are given a positive integer N.

Construct a **minimal** length string S consisting of letters `'u'` and `'w'` only, satisfying the following condition:

- The number of subsequences of S which equal `"uwu"` is exactly N. Note that a subsequence need not be continuous.

It can be proven that at least one valid string exists. You have to find the shortest possible string for each test. The input is generated in such a way that the sum of the lengths of the minimal strings is at most 10^7 over all test cases.

Input Format

- The first line of input contains a single integer T, denoting the number of test cases.
- The first and only line of each test case contains N — the required number of subsequences.

Output Format

For each test case, output a **minimal** length string S with exactly N `"uwu"` subsequences.

Constraints

- $1 \leq T \leq 10^3$
- $1 \leq N \leq 10^{18}$
- The sum of the minimum lengths of the required strings does not exceed 10^7.

Samples

Sample Input 1

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

Sample Output 1

```
uwu
uwwu
uwwwu
uwwuu
uwwwwwu
uwwwuu
```

Sample Explanation

Test Case 1: `"uwu"` has exactly one subsequence which equals `"uwu"`, which is the whole string itself.
Test Case 2: `"uwwu"` has 2 subsequences which equal `"uwu"`, using 1-based indexing:

- The subsequence formed by indices $(1, 2, 4)$.
- The subsequence formed by indices $(1, 3, 4)$.

Problem B
Counting Distance Arrays

Time limit: 2 seconds
Memory limit: 1.5 Gigabytes

Problem Description

Given a tree with N nodes numbered 1 to N, and a subset of nodes S, we define an array A of size N in the following way:

- $A_i = \max_{x \in S}(\text{dist}(x, i))$.

Here, $\text{dist}(x, y)$ represents the number of edges on the unique shortest path between x and y.

There are $2^N - 1$ non-empty subsets S possible. Find the number of possible distinct arrays A over all $2^N - 1$ choices of S. Since the answer may be large, output it modulo 998244353.

Input Format

- The first line of input contains 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 — the number of nodes in the tree.
  - The next $N - 1$ lines each contain 2 integers u and v — representing an edge (u, v) in the tree.

Output Format

For each test case, output on a new line the number of distinct arrays A possible over all $2^N - 1$ non-empty subsets S, modulo 998244353.

Constraints

- $1 \leq T \leq 10^4$
- $2 \leq N \leq 2 \cdot 10^5$
- $1 \leq u, v \leq N$
- The set of $N - 1$ input edges form a tree.
- The sum of N does not exceed $2 \cdot 10^5$ over all test cases.

Samples

Sample Input 1

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

Sample Output 1

```
3
6
23
```

Sample Explanation

Test Case 1: There are 3 subsets S possible. Each of them produces a distinct result, enumerated as follows:

- $S = \{1\}$, the array $A = [0, 1]$
- $S = \{2\}$, the array $A = [1, 0]$
- $S = \{1, 2\}$, the array $A = [1, 1]$

Problem C
Counting is Fun

Time limit: 2 seconds
Memory limit: 1.5 Gigabytes

Problem Description

You are given an integer S.

Let $f(N)$ denote the sum of **inversion numbers**[†] for all arrays A with positive integers, and of length N with sum S.

Formally, $f(N)$ is the sum of **inversion numbers** for arrays satisfying the following conditions:

- $|A| = N$
- $1 \leq A_i \leq S$ and A_i are integers
- $\sum_{i=1}^{N} A_i = S$

Find the values of $f(1), f(2), \ldots, f(S)$ modulo 998244353.

[†] The **inversion number** of an array A is the number of pairs satisfying $1 \leq i < j \leq |A|$ and $A_i > A_j$.

Input Format

The first and only line of input contains a single integer S.

Output Format

Print S integers, the values of $f(1), f(2), ..., f(S)$ modulo 998244353.

Constraints

$2 \leq S \leq 2 \cdot 10^5$

Samples

Sample Input 1

```
4
```

Sample Output 1

```
0 1 3 0
```

Sample Input 2

```
10
```

Sample Output 2

```
0 4 48 204 460 600 462 196 36 0
```

Sample Explanation

Test case 1: The answers are computed as below:

- $N = 1$: There is only one array $[4]$. This has no inversions.
- $N = 2$: There are 3 arrays, $[1, 3]$, $[2, 2]$ and $[3, 1]$. There is 1 inversion in the first array, and 0 in the others.
- $N = 3$: There are 3 arrays, $[1, 1, 2]$, $[1, 2, 1]$ and $[2, 1, 1]$. The inversion numbers are $0, 1, 2$ respectively. Hence, the sum is 3.
- $N = 4$: Only one array $[1, 1, 1, 1]$. This has no inversions.

Problem D
Deletable Subarrays

Time limit: 1 second
Memory limit: 1.5 Gigabytes

Problem Description

An array B is called *deletable* if and only if it can be made completely empty by using the following operation several times:

- Choose $1 \leq i < |B|$ such that $B_i = B_{i+1}$, and remove both B_i and B_{i+1} from the array.

Given an array A of length N, count the number of *deletable* subarrays. Formally, compute the following sum:

$$\sum_{L=1}^{N} \sum_{R=L}^{N} [\text{Is } A[L, R] \text{ deletable?}]$$

where $A[L, R]$ represents the subarray $[A_L, A_{L+1}, \ldots, A_R]$.

Input Format

- The first line of input contains 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 one integer N.
  - 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 number of *deletable* subarrays of A.

Constraints

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

Samples

Sample Input 1

```
3
4
1 2 2 1
4
1 1 1 1
12
12 7 2 3 2 2 3 2 7 1 1 12
```

Sample Output 1

```
2
4
7
```

Sample Explanation

Test Case 1: The subarrays $[1, 2, 2, 1]$ and $[2, 2]$ are *deletable*.

Test Case 2: All even-length subarrays are *deletable*.

Problem E
Greedy Prices

Time limit: 2 seconds
Memory limit: 1.5 Gigabytes

Problem Description

There are N items for sale at an auction. However, the auction is greedy and has made the prices variable depending on how much it perceives you will be able to pay. If you have already spent some amount of money, then it will increase the cost proportionally, as you are more likely to be willing to buy the item even at a higher price.

Formally, the i-th item has 2 parameters M_i and C_i, and the cost of the item is $M_i \cdot X + C_i$, where X is the **amount of money** you have already spent.
Each item can be bought at most once. You can decide the order of buying items.

You have to answer Q queries of the following form:

- If you had a budget of P_i, what is the maximum number of items you can buy?

Input Format

- The first line of input contains 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 2 integers — N and Q, the number of items and the number of queries.
  - The second line contains N integers — $M_1, M_2, \ldots, M_N$, the linear coefficients of the items.
  - The third line contains N integers — $C_1, C_2, \ldots, C_N$, the constant coefficients of the items.
  - The next Q lines each contain 1 integer — P_i, representing a budget query.

Output Format

For each test case, output Q integers - the answer to the queries in order.

Constraints

- $1 \leq T \leq 10^4$
- $1 \leq N, Q \leq 2 \cdot 10^5$
- $0 \leq M_i, C_i, P_i \leq 10^9$
- The sum of N and the sum of Q both do not exceed $2 \cdot 10^5$ over all test cases.

Samples

Sample Input 1

```
2
3 4
1 1000 0
3 6 4
2
6
7
19
6 6
0 0 1 1 2 3
0 1 0 2 3 4
0
1
8
15
3
10000
```

Sample Output 1

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

Sample Explanation

Test Case 1: Here are the answers to the respective queries:

- Budget 2: Impossible to buy any item.
- Budget 6: We can buy any of the 3 items for the prices $3, 6, 4$ respectively. However, we cannot buy more than 1 item.
- Budget 7: First, buy the 1^{st} item for a cost of 3, and then buy the 2^{nd} item for 4.
- Budget 19: Buy all 3 of the items in the following order:
  - $X = 0$, Buy item 2 for a cost of 6.
  - $X = 6$, Buy item 1 for a cost of $1 \cdot 6 + 3 = 9$.
  - $X = 15$, Buy item 3 for a cost of $0 \cdot 15 + 4$.
  - We bought all 3 items spending exactly 19.

Problem F
Jagged Operations

Time limit: 1 second
Memory limit: 1.5 Gigabytes

Problem Description

An array A of size N is called **jagged** if and only if there exists some triplet of indices i, j, and k ($1 \leq i < j < k \leq N$) such that $A_i > A_j$ and $A_j < A_k$.

You are given an array A and you want to make it **not jagged**. To do this, you can perform the following operation as many times as needed:

- Choose L, R, X such that $1 \leq L \leq R \leq N$ and X is an integer, then add X to all indices between L and R.

  Formally, set $A_i \leftarrow A_i + X$ for all $L \leq i \leq R$.

Find the minimum number of operations needed.

Input Format

- The first line of input contains 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 size of the array.
  - 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 minimum number of operations needed.

Constraints

- $1 \leq T \leq 10^4$
- $3 \leq N \leq 2 \cdot 10^5$
- $1 \leq A_i \leq 10^9$
- The sum of N does not exceed $2 \cdot 10^5$ over all test cases.

Samples

Sample Input 1

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

Sample Output 1

```
0
1
```

Sample Explanation

Test Case 1: The array is already **not jagged**.

Test Case 2: The initial array is jagged as $(i, j, k) = (1, 2, 3)$ satisfies the conditions.
We can choose $L = 2, R = 2, X = 1$, and modify the array to $[2, 2, 2]$. This is **not jagged**.

Problem G
LCM on Range

Time limit: 4 seconds
Memory limit: 1.5 Gigabytes

Problem Description

Given an integer N, you need to find a range $[L, R]$ satisfying the following conditions:

- $1 \leq L < R \leq 10^{18}$
- $\text{LCM}(L, L+1, L+2, \ldots, R-1, R) = N$, where LCM represents the least common multiple.

If no such range exists, output -1 instead.

Input Format

- The first line of input contains a single integer T, denoting the number of test cases.
- The first and only line of each test case contains N — the target value.

Output Format

For each test case, output the following:

- If a valid range L, R exists, output the integers L and R satisfying $1 \leq L < R \leq 10^{18}$.
- Otherwise, output -1.

Constraints

- $1 \leq T \leq 10^5$
- $1 \leq N \leq 10^{18}$

Samples

Sample Input 1

```
4
1
2
6
12
```

Sample Output 1

```
-1
1 2
1 3
1 4
```

Sample Explanation

Test Case 1: There exists no valid range. Note that $L = R = 1$ is not valid as we need $L < R$.

Test Case 2: $\text{LCM}(1, 2) = 2$.

Problem H
Majority Graph

Time limit: 5 seconds
Memory limit: 1.5 Gigabytes

Problem Description

You are given an array A with N elements - $A_1, A_2, \ldots, A_N$.

Construct a graph G on N nodes in the following way:

- Draw an edge (i, j) if and only if the following conditions are satisfied:
  - $1 \leq i < j \leq N$
  - The subarray $A[i, j]$ has a **majority element**.

Find the number of connected components of G.

The subarray $A[i, j]$ refers to the array $[A_i, A_{i+1}, \ldots, A_j]$.
An array B of length M has a **majority element** if and only if there is some element X that occurs **strictly more** than $\frac{M}{2}$ times in B.

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 a single integer N - the size of the array and graph.
  - 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 number of connected components of the graph G.

Constraints

- $1 \leq T \leq 10^5$
- $2 \leq N \leq 2 \cdot 10^6$
- $1 \leq A_i \leq N$
- The sum of N over all test cases does not exceed $2 \cdot 10^6$.
- Note that the **constraint on N is higher than usual**.

Samples

Sample Input 1

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

Sample Output 1

```
2
4
1
1
```

Sample Explanation

Test Case 1: There are 2 pairs (i, j) that satisfy the majority condition: $(1, 3)$ and $(2, 4)$. The subarray $A[1, 3]$ has 1 as a majority element, and $A[2, 4]$ has 2. Thus, the graph has 2 connected components.

Test Case 2: There is only one edge $(2, 4)$. Hence, there are 4 connected components.

Problem I
Suffix Array

Time limit: 1 second
Memory limit: 1.5 Gigabytes

Problem Description

You are given a string S of length N.

The suffix strings of S are the non-empty strings which can be formed by deleting some (possibly 0) characters from the beginning of S, i.e. for each i ($1 \leq i \leq N$), the substring $S_i S_{i+1} \ldots S_N$ is a suffix of S.

There are $N - 1$ suffix strings of S that have length ≥ 2.
Your task is to sort these suffix strings of S in lexicographic increasing[†] order and then print the 2^{nd} characters of each of the strings.

[†] A string A is lexicographically smaller than B if any of these conditions hold:

- There exists i, $1 \leq i \leq \min(|A|, |B|)$ such that $A_i < B_i$, and $A_j = B_j$ for all $1 \leq j < i$.
- $|A| < |B|$ and for all $1 \leq i \leq |A|$, $A_i = B_i$.

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 a single integer N - the length of the string.
  - The second line contains a string S of length N.

Output Format

For each test case, output the 2^{nd} characters of the sorted suffix strings of S (without spaces between them).

Constraints

- $1 \leq T \leq 10^4$
- $2 \leq N \leq 2 \cdot 10^5$
- $|S| = N$
- S only contains lowercase Latin characters.
- The sum of N over all test cases does not exceed $2 \cdot 10^5$.

Samples

Sample Input 1

```
3
4
asia
4
west
6
abbaab
```

Sample Output 1

```
sai
ste
abbab
```

Sample Explanation

Test Case 1: There are 3 suffix strings of S with length ≥ 2, namely asia, sia, and ia.

The sorted order is asia, ia, sia. The 2^{nd} characters in this order are s, a, and i. Hence, the output is sai.

Problem J
Tri-Knight

Time limit: 1 second
Memory limit: 1.5 Gigabytes

Problem Description

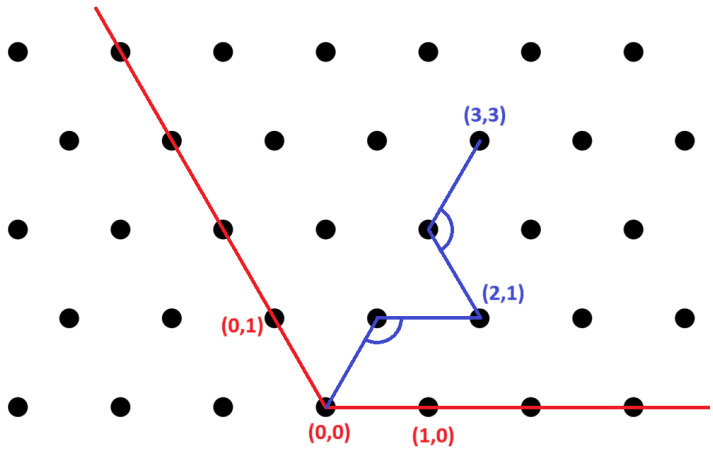
Have you ever imagined how a chess knight will move on a triangular grid? A triangular grid is a grid where each point has 6 neighbors, formed in 60° angles. The coordinates are written with respect to two axes, and in this problem, the two axes have an inner angle of 120°.

(X, Y) is the point that is reached by taking X steps of unit length along the horizontal axis, and then Y steps along the oblique axis. Please check the diagram.

The points adjacent to (X, Y) are $(X+1, Y)$, $(X+1, Y+1)$, $(X, Y+1)$, $(X-1, Y)$, $(X-1, Y-1)$, and $(X, Y-1)$.

On the grid, a **tri-knight**'s movement on one point A is defined as follows.

- First, it selects a point B adjacent to A.
- Then, it moves to a point C adjacent to B, so that the angle $\angle ABC$ is **obtuse**.



The above picture shows the grid and the coordinate system, and how one can move to $(3, 3)$ in 2 moves. Note that the grid extends infinitely in all directions (including negative).

Given a point (X, Y) on an infinite triangular grid, please find the **minimum number of movements** required for a tri-knight to move from $(0, 0)$ to (X, Y). If the tri-knight is never able to visit the given point, report -1.

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, X and Y — representing the coordinates in the triangular grid system.

Output Format

For each test case, output the following:

- If the tri-knight can visit (X, Y), output the minimum moves it needs.
- Otherwise, output -1.

Constraints

- $1 \leq T \leq 10^4$
- $-10^9 \leq X, Y \leq 10^9$

Samples

Sample Input 1

```
5
3 3
1 1
1 -1
-3 -3
-2 998244353
```

Sample Output 1

```
2
-1
1
2
665496236
```

Sample Explanation

Test Case 1: As written in the statement, the tri-knight can reach $(3, 3)$ in 2 moves by going to $(2, 1)$ and then $(3, 3)$.

Test Case 2: It can be proven that it is impossible to reach $(1, 1)$.

Test Case 3: The tri-knight can reach the point $(1, -1)$ in 1 move. For this, $A = (0, 0)$ initially, it chooses B (adjacent to A) as $(-1, 0)$ and then the point C (adjacent to B) as $(-1, 1)$. Note that here, $\angle ABC = 120^\circ$ which is obtuse.

Problem K
Unique Subsequences Counting

Time limit: 1 second
Memory limit: 1.5 Gigabytes

Problem Description

An array A of length N is said to be *good* if and only if every subsequence of length M appears at most once in the array.

For example, $A = [1, 2, 2]$ is not *good* with length parameter $M = 2$, because the subsequence $[1, 2]$ appears twice. On the other hand, it would be *good* with length parameter $M = 3$.

Given N and M, count the number of arrays A satisfying the following conditions:

- $|A| = N$.
- $1 \leq A_i \leq N$.
- A is *good*.

Since the answer may be large, output it modulo $10^9 + 8$. Please note the **unusual modulo**.

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 N and M.

Output Format

For each test case, output on a new line the number of sequences satisfying the above constraints modulo $10^9 + 8$.

Constraints

- $1 \leq T \leq 10^4$
- $2 \leq M \leq N \leq 10^9$

Samples

Sample Input 1

```
2
2 2
50 25
```

Sample Output 1

```
4
837455472
```

Sample Explanation

Test Case 1: There are 4 arrays satisfying all conditions:

- $[1, 1]$
- $[1, 2]$
- $[2, 1]$
- $[2, 2]$