Midterm 2 - ICT20211 - 0606

A. K-Consecutive Sum

1 second, 256 megabytes

Given 2 numbers x, k. Determine whether or not there exists a sequence of k consecutive numbers with the sum equal to x.

For example, let $x = 10, k = 4$, then the sequence is $[1, 2, 3, 4]$.

Input

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

Each test case consists of a single line that contains two integers x and k $(|x| \le 10^{16}, 1 \le k \le 10^{18})$.

Output

For each test case, output on a separate line: YES if the answer is "Yes", NO if the answer is "No".

input
4 10 4 15 4 7 7 -6 3
output
YES NO YES YES

Subtask 1 (50%): $T \le 10, |x| \le 1000, 1 \le k \le 1000$.

Subtask 2 (50%): No more addition constraint.

B. Longest Path

1 second, 256 megabytes

You are given a weighted undirected connected tree that contains n nodes and $n - 1$ edges. Every edge has a weight and there is exactly one path between every pair of different nodes. Find the longest path in the tree.

Input

The first line contains one integer $n (2 \le n \le 200000)$

The next $n - 1$ lines will contain 3 integers for each, $u, v, w (1 \le u, v \le n, w \le 10^9)$, which means that there is an edge of weight w connecting the nodes u and v.

It is guaranteed that the edges describe one connected tree.

Output

The value of the longest path in the tree.

input
7 1 2 1 1 3 7 2 6 3 2 7 -2 7 4 4 7 5 6
output
12

Subtask 1 (50%): $n \le 1000$.

Subtask 2 (50%): No more addition constraint.

C. Maximum Partition

1 second, 256 megabytes

Given an array A of N integers. We define the value of a subarray $a_i, a_{i+1}, a_{i+2}, \ldots, a_{r-1}, a_r (l \leq r)$ as the minimum element in the subarray. Find the maximum sum of all k partition of the array.

As an example, consider the array $[9, 5, 10, 7, 9, 5]$. The partition formed by the subarrays $[9, 5]$, $[10]$ and $[7, 9, 5]$ has cost $5 + 10 + 5 = 20$, while the partition formed by the subarrays $[9]$, $[5, 10, 7]$ and $[9, 5]$ has cost $9 + 5 + 5 = 19$. Both partitions are formed by $k = 3$ subarrays, but they have different costs. Other partitions may have different costs.

Input

The first line contains an integer $N (1 \leq N \leq 8000)$, the number of elements in the array A.

The second line contains N integers $a_1, a_2, \ldots, a_N (1 \leq a_i \leq 10^9$ for $i = 1, 2, \ldots, N)$ representing the array.

The third line contains a single integer $T (T \leq N)$.

The next T lines contains a single integer $k (k \leq N)$.

Output

Consist of T lines, each line contains a single integer - the maximum value of k partition of the array.

input
6 9 5 10 7 9 5 3 1 3 2
output
5 20 14

Specification for the example:

- $k = 1 \Rightarrow [9, 5, 10, 7, 9, 5]$
- $k = 2 \Rightarrow [9], [5, 10, 7, 9, 5]$
- $k = 3 \Rightarrow [9, 5], [10], [7, 9, 5]$

Subtask 1(40%): $N \leq 20$.

Subtask 2(40%): $N \leq 100$.

Subtask 3(20%): No more addition constraints.