Problem Set

Softcom'23 Speed Programming

November 18' 2023.

Instructions

- Do not open the booklet unless you are explicitly told to do so. You can only read these instructions below.
- Do not create disturbance or move around unnecessarily in the arena.
- If you have any question regarding the problems, send a clarification from the judges using DOMJudge.
- There would be no internet access and mobile phones are also not allowed.
- Before submitting a run, make sure that it is executable via command line. For Java, it must be executable via "javac" and for GNU C++ via "g++". Java programmers need to remove any "package" statements and source code's file name must be the same as of main class. C++ programmers need to remove any getch() / system("pause") like statements.
- Do not attach input files while submitting a run, only submit/attach source code files, i.e., *.java or *.cpp or *.py.
- Language supported: C/C++, Java and Python3
- Source code file name should not contain white space or special characters.
- You must take input from Console i.e.: Standard Input Stream (stdin in C, cin in C++, System.in in Java, stdin in Python)
- You must print your output to Console i.e.: Standard Output Stream (stdout in C, cout in C++, System.out in Java)
- Please, don't create/open any file for input or output.
- Please strictly meet the output format requirements as described in problem statements, because your program will be auto judged by computer. Your output will be compared with judge's output byte-by-byte and not tolerate even a difference of single byte. So, be aware! **Pay special attention to spaces, commas, dots, newlines, decimal places, case sensitivity etc.**
- Unless mentioned in some problem, all your programs must meet the time constraint of 5 seconds.
- The decision of judges will be absolutely final.

Question 01: Power Of Two.

Time limit: 1 second

Ahmed is fascinated by the power of 2 and wants to create a program to check whether a given positive integer is a power of 2 or not. A number is considered a power of 2 if it can be expressed in the form 2^k where k is a non-negative integer. For example, if Ahmed's function is called with n = 16, it should print YES since 16 is 2^4 and if called with n = 5, if should print NO

Input Format:

- The first line contains an integer t ($1 <= t <= 250$), the number of test cases.
- The next line contains number n ($0 <= n <= 2^{32}-1$) to be checked whether it is of power 2 or not.

Output:

For each test case, print YES number is in power of 2. Otherwise, print NO.

Sample Input	Sample Output
5	YES
4	NO
17	NO
30	YES
32	YES
512	

Question 02: Friendly Aliens Come to GIK Institute.

Time limit: 3 seconds

GIK Institute is located at the base of the scenic Tarbela dam, surrounded by picturesque natural beauty and breathtaking views. The dam itself is a stunning landmark and a source of pride for the university and the local community. The surrounding area is lush with greenery and dotted with trees, creating a peaceful and serene environment for the university campus. The body of water created by the dam adds to the overall aesthetic appeal of the university, and offers opportunities for recreational activities such as boating or fishing. Additionally, the university has walkways, parks, or green spaces near the water where students and faculty can gather and enjoy the scenic views. The close proximity to the dam and the water makes the university a unique and attractive place to study, work, and live. The combination of the stunning natural surroundings, academic opportunities, and close-knit community makes GIKI a truly special place. A few years ago, a friendly group of aliens from a distant planet landed on a mountain range. They have a favorite food which they bring with them called Kiti-Kata. When a group of people approach them, they choose one person to share their favorite food with. To determine the chosen person, all members of the group stand in a circle. They number the persons 1 to p. Aliens follow an elimination process to choose a person, stated below:

- In the first step, person 1 is removed.
- In the second step, they skip one person. Person 3 is removed.
- In the third step, they skip two persons. Person 6 is removed.
- And so on...
- The last person in the circle is the chosen one.

For instance, if a group of five people visited the aliens, they would be removed in this order: 1, 3, 2, 5, and the person at position 4 would eat kiti-kata with the aliens. Similarly, if a group of four approached, they would be removed in this sequence: 1, 3, 4, and the person at position 2 would be treated to the alien's hospitality. Can you help identify the lucky person to eat kiti-kata during the next trip of aliens to the mountains?

Input Format:

- The first line contains an integer t, the number of test cases ($1 <= t <= 1000$).
- The next line contains total number of people p ($1 <= p <= 5000$).

Output:

The output of the program contains n ($1 \le n \le 1000$) lines, where each line consists of a single number, indicating the position of the person to eat kiti-kata.

Sample Input	Sample Output
5	4
5	4
7	8
12	11
15	13
25	

Question 03: Sorting Permutation with Rotations.

Time limit: 5 seconds

Sarah has been given an array A representing a permutation of a sequence of natural numbers starting from 1. Her goal is to determine whether it's possible to achieve a sorted array using a specific operation. This operation allows the rotation of elements in any set of consecutive indices, following the pattern: **XYZ → YZX → ZXY → XYZ**. For each test case, print "YES" if the array can be fully sorted using this operation, and print "NO" otherwise.

Input Format:

- The first line contains an integer *t* (1 <= *t* <= 1000), the number of test cases.
- The next t pairs of lines are as follows:
- The first line contains an integer *n* (3 ≤ *n* ≤ 1000), the length of *A*.
- The next line contains *n* space-separated integers *A[i]*.

Output:

The output of the program contains *n* (1 ≤ *n* ≤ 1000) lines where each line consists of YES if *A* can be fully sorted else NO.

Sample Input	Sample Output
3 3 3 1 2 4 1 3 4 2 5 1 2 3 5 4	YES YES NO

Explanation:

Test Case 0:
A = {3, 1, 2} → rotate A(3, 1, 2) → A(1, 2, 3) is now sorted, so we print YES on a new line.

Test Case 1:
A = {1, 4, 3, 2} → rotate A(3, 4, 2) → A ={1, 4, 2, 3}, rotate A(4, 2, 3) → A(1, 2, 3, 4) is now sorted, so we print YES on a new line.

Test Case 2:
A = {1, 2, 3, 5, 4} → rotate A(3, 5, 4) → rotate A(5, 4, 3) → rotate A(4, 3, 5). No sequence of rotations will result in a sorted . Thus, we print NO on a new line.

Question 04: Detective Harris' String Conundrum: Unraveling the Code of Validity.

Time limit: 5 seconds

Harris, a brilliant student at Lahore University of Management Sciences (LUMS) and a part-time seasoned detective with a keen eye for patterns in strings, has formulated an intriguing criterion for determining the validity of a string. He considers a string to be valid under two conditions:

1. **Uniform Frequency**: All characters in the string must appear the same number of times.
2. **Selective Removal**: Alternatively, the string is also valid if Harris can strategically remove just one character at a specific index, and the remaining characters will then occur the same number of times.

Harris believes that unraveling the mysteries of strings can lead to deeper insights. Your mission is to aid Detective Harris by taking a string s as input and returns "YES" if the string is valid according to his criteria, and "NO" otherwise. For instance:

- s = abc, this is a valid string because frequencies are {a: 1, b: 1, c: 1}.
- s = abcc, this is a valid string because we can remove one c and have 1 of each character in the remaining string {a: 1, b: 1, c: 1}.
- s = abccc, this string is not valid as we can only remove 1 occurrence of c. That leaves character frequencies of {a: 1, b: 1, c: 2}.

Input Format:

- The first line contains an integer t ($1 <= t <= 1000$), the number of test cases.
- The next line contains string s ($1 <= |s| <= 10^5$). $s[i]$ belongs to ascii[a-z].

Output:

For each test case, print YES if string is valid. Otherwise, print NO.

Sample Input	Sample Output
5 abcc abccc aabbcd abcdefghhgfedecba aabbccddeefghi	YES NO NO YES NO

Question 05: Crafting Equity in Peshawar.

Time limit: 3 seconds

In the lively city of Peshawar, there was a smart guy named Akram. He loved solving tricky math problems and exploring the streets of his hometown. One day, while working on a tough puzzle, he found an old book with a special trick inside. This trick was about making a group of numbers in a way that made the numbers fair to each other. It meant the equity between the biggest and smallest numbers in the group shouldn't be too big. The trick asked Akram to use a list of numbers, which he called arr, and another number called k. His job was to create a new group of numbers, let's call it arr', from the original list in a way that made the equity as small as possible. **Equity** is just a simple way of saying the difference between the biggest and smallest numbers in the group. Akram wanted this difference to be as small as possible. In the original array, numbers could be repeated, and that made things interesting as they could be used in subarray as many times they appear in original array. Akram realized he needed to figure out the best way to arrange the numbers to make the new group as fair as possible. As Akram started working on this challenge, he walked through the familiar streets of Peshawar. Trying different combinations of numbers, he aimed to find the one that made the equity between the biggest and smallest numbers as minimum as possible. Trials and errors marked the pages of his journey. Each attempt brought him closer to understanding the intricate dance of numbers that could create an equitable array. The streets of Peshawar became his muse, inspiring the arrangement of elements in ways that mirrored the city's cultural richness.

You will be provided with a list of integers, arr, and a single integer, k. Your task is to form an array, arr', with a length of k using elements from arr in such a way that the array's equity is minimized. The equity of an array is determined by the calculation:

Equity = **max(arr') – min(arr'),**

where, max is largest in arr' and min is the smallest in arr'. The goal is to craft an array that achieves the smallest *possible difference* between its maximum and minimum values, ensuring a minimal equity. Consider the *following example below:*

- arr = [1, 4, 7, 2], k = 2
- Pick any two elements, say arr'[4, 7]. Equity = max(4, 7) – min(4, 7) = 3
- Testing for all pairs, the solution [1, 2] provides the minimum Equity, 1.
- Note: Integers in array may ***not*** be unique.

Input Format:

- *The first line* contains an integer *t* (1 <= *t* <= 1000), the number of test cases.
- *The next line* contains an integer *n* (2 <= *n* <= 10^5), the number of elements in array.
- *The next line* contains an integer *k* (2 <= *k* <= *n*), *k* length array to calculate unfairness.
- Each of the next lines contains an integer *arr[i]* where 0 <= i <= n.

Output:

For each test case, return int: the minimum possible equity.

Sample Input	Sample Output
2 7 3 10 100 300 200 1000 20 30 5 2 1 2 1 2 1	Case #1: 20 Case #2: 0