A. Challenge Accepted

1 second, 256 megabytes

Odyssey has concluded, but no one can forget it, particularly the performance by Shankar–Ehsaan–Loy. All ADA students are feeling melancholic. Udit, the most generous individual in our college, has come up with a challenge to reignite the enthusiasm for ADA among students. Udit always remains alone due to his obsession with subsequences, and his challenge is evidence of it.

You are given an array with m elements. He asks you to determine whether a subsequence exists in that array where the sum of elements of that subsequence is a multiple of n. However, as you all know, Udit is the most generous person in our college. He **removes any one element** from the array that you don't know. Your task is to check whether the remaining array formed by removing any one element has a subsequence whose **sum is a multiple of** n (sarcasm spotted). Here, the **selected subsequence must contain at least one element**.

A subsequence of an array A, denoted as B, can have none, some, or all elements of array A. For example, if A={1,2,3}, then its subsequences are {},{1},{2},{3},{1,2},{2,3},{1,3}, and {1,2,3}.(According to the question, null subsequence is not allowed to be used).

If a subsequence exists from the remaining array after removing any one element whose sum is divisible by n, then print "YES"; otherwise, print "NO".

Input

The first line contains one integer t $(1 \leq t \leq 5)$ — the number of test cases. Each test case consists of two lines.

The first line contains two integers m $(1 \leq m \leq 18)$ — the length of the array and n $(1 \leq n \leq 10^9)$.

The second line contains m integers $a_1, a_2, a_3 \ldots . a_m$ $(-10^9 \leq a_i \leq 10^9)$.

It is guaranteed that the sum of m over all test cases does not exceed 18.

Output

Print "YES" if you can find such a subsequence, "NO" otherwise.

input
1 5 1 1 2 3 4 5
output
YES

For 1st test case:-

There are 5 possible cases:

1. If remove 1, A_remaining=[2,3,4,5]

- any subsequence sum will work as every number is divisible by 1

2. If remove 2, A_remaining=[1,3,4,5]

- any subsequence sum will work as every number is divisible by 1

3. If remove 3, A_remaining=[1,2,4,5]

- any subsequence sum will work as every number is divisible by 1

4. If remove 4, A_remaining=[1,2,3,5]

- any subsequence sum will work as every number is divisible by 1

5. If remove 5, A_remaining=[1,2,3,4]

- any subsequence sum will work as every number is divisible by 1

B. THE BATTLE OF HOGWARTS

1 second, 256 megabytes

The battle of Hogwarts has finally begun, and it is Harry's team vs Voldemort's army.

Voldemort has divided his army into b bases (spread throughout Hogwarts). Each base has a defensive power d_i, and w_i innocent wizards which it is keeping hostage and whom they threaten to kill, if Harry doesn't surrender.

Seeing this, Harry also decided to divide his team into n teams, each with a certain attacking power a_i. A team can attack all bases which have a defensive power less than or equal to its attacking power. If a team attacks a base it saves all the wizards in that base.

Harry is still undecided which team to send out first, so he would like to know for each team what is the maximum amount of wizards it can save.(independently of all others).

Input

The first line contains integers n and b $(1 \leq n, b \leq 10^5)$, the number of teams Harry created and the number of Voldemort's bases, respectively.

The second line contains n integers a_i $(0 \leq a_i \leq 10^9)$, the attacking power of each team.

The next b lines contain integers d_i, w_i $(0 \leq d_i \leq 10^9, 0 \leq w_i \leq 10^4)$, the defensive power and the wizards at each base, respectively.

Output

Print n integers, the maximum amount of wizards each team can save, in the same order as the teams are given in the input.

input
5 4 1 3 5 2 4 0 1 4 2 2 8 9 4
output
1 9 11 9 11

Explanation of the first testcase:

The first team can only attack the first base.

The second team can attack the first and third bases.

The third team can attack the first, second and third bases $(1 + 2 + 8 = 11)$ and so on.

C. RPG BUT WITH STRINGS

2 seconds, 256 megabytes

Welcome to the RPG Quest for the Perfect String! You are the lone hero who has survived the waves of DSA on the island of C. Now, on a quest to save the world, you come across a dungeon of strings. A string is considered to be perfect for character $char$ if either of these conditions are true:

1. The length of the string is 1, and it has the required character $char$.
2. Length isn't 1, but the first half is perfect for the character $char + 1$ only, while the second is filled with character $char$ only.
3. Length isn't 1, but the first half is filled with character $char$ only, while the second half is perfect for the character $char + 1$.

Now, guessing whether a string is perfect or not is just too easy for you, so you decide to make it more challenging for yourself. You will have a certain amount of energy initially, and an energy unit is used to change a character at any place into any other character at any instant.

You have to try and tell if you can convert the string into a perfect string for the character 'a'. If you can do the task, print the minimum energy required to do it; otherwise, print "-1".

Note- There is no character after 'z'. So there is no ('z'+1). Also, it is given that all the strings are in lowercase alphabets.

Input

The first line of the input contains one integer $t (1 \le t \le 2 \cdot 10^4)$ — the number of test cases. Then t test cases follow.

The first line of the test case contains one integer $n (1 \le n \le 131\,072)$ — the length of s. It is guaranteed that $n = 2^k$ for some integer $k \ge 0$. The second line of the test case contains the string s consisting of n lowercase Latin letters.

It is guaranteed that the sum of n does not exceed $2 \cdot 10^5$ $\sum n \le 2 \cdot 10^5$

Output

For each test case, if it is possible to do it, print the minimum amount of energy required. Otherwise print "-1".

input
2 8 cdbbaaaa 8 ceaaaabb

output
0 4

D. Pokemon Master Quest

2 seconds, 256 megabytes

Ash has set out on his journey to become a pokemon master. To choose his starter pokemon he is currently standing in professor Oak's lab . Professor Oak has n pokemon (all standing next to each other) for Ash to choose from . Ash can choose any number of pokemon to start with, however he wants to choose only that subset which is most friendly.

Each pokemon has a friendliness stat represented by a_i. To help him, Professor provides him with a list of q numbers, the i^{th} of which is the total friendliness stat choosen by the greatest pokemon trainers of history when they started out on their journey.

For all q numbers Ash wants to know whether he can choose some pokemon with total stat = s_i. However there is a condition, he can only choose pokemon in the following way :-

- Assume $mid = \frac{\lfloor max(array) + min(array) \rfloor}{2}$, where max and min — are functions that find the maximum and the minimum stats . In other words, mid is the sum of the maximum and the minimum friendliness stat divided by 2 rounded down.
- Then the array of pokemon is split into two parts $left$ and $right$. The $left$ array contains all Pokemon which have stat less than or equal mid, and the $right$ array contains all pokemon which have stat greater than mid. Pokemon in $left$ and $right$ keep their relative order from array.
- During the third step we choose which of the $left$ and $right$ arrays we want to keep. The chosen array replaces the current one, and the pokemon in the other array get angry and run away.

These set of operations forms a procedure which can be repeated any no. of times (possibly 0)

You need Ash to find the result of all q tests. For each test, he starts again with the same set of n Pokemon.

Input

Each test contains one or more test cases. The first line contains the number of test cases t $(1 \le t \le 100)$.

The first line of each test case contains two integers n and q $(1 \le n, q \le 10^5)$ — the length of the array a and the list of numbers provided by Professor Oak.

The second line of each test case contains n integers $a_1, a_2, \ldots, a_n$ $(1 \le a_i \le 10^6)$ — the friendliness stats of the pokemon.

Next q lines of each test case contain a single integer s_i $(1 \le s_i \le 10^9)$ — the sum of friendliness stats which Ash wants to get in the i^{th} test.

It is guaranteed that the sum of n and the sum of q does not exceed 10^5 $(\sum n, \sum q \le 10^5)$.

Output

Print q lines, each containing either a "Yes" if the corresponding sum of friendliness stat can be achieved and "No" in the opposite case.

input
2 5 5 1 2 3 4 5 1 8 9 12 6 5 5 3 1 3 1 3 1 2 3 9 11

output
Yes No Yes No Yes No Yes No Yes Yes

Explanation of the first test case:

1. We can get an array with the sum $s_1 = 1$ in the following way:
   a. $a = [1, 2, 3, 4, 5]$, $mid = \frac{1+5}{2} = 3$, $left = [1, 2, 3]$, $right = [4, 5]$. We choose to keep the $left$ array.
   b. $a = [1, 2, 3]$, $mid = \frac{1+3}{2} = 2$, $left = [1, 2]$, $right = [3]$. We choose to keep the $left$ array.
   c. $a = [1, 2]$, $mid = \frac{1+2}{2} = 1$, $left = [1]$, $right = [2]$. We choose to keep the $left$ array with the sum equalling 1.
2. It can be demonstrated that an array with the sum $s_2 = 8$ is impossible to generate.
3. An array with the sum $s_3 = 9$ can be generated in the following way:
   a. $a = [1, 2, 3, 4, 5]$, $mid = \frac{1+5}{2} = 3$, $left = [1, 2, 3]$, $right = [4, 5]$. We choose to keep the $right$ array with the sum equalling 9.
4. It can be demonstrated that an array with the sum $s_4 = 12$ is impossible to generate. We can get an array with the sum $s_5 = 6$ in the following way:
   a. $a = [1, 2, 3, 4, 5]$, $mid = \frac{1+5}{2} = 3$, $left = [1, 2, 3]$, $right = [4, 5]$. We choose to keep the $left$ with the sum equalling 6.
5. We can get an array with the sum $s_5 = 6$ in the following way:
   a. $a = [1, 2, 3, 4, 5]$, $mid = \frac{1+5}{2} = 3$, $left = [1, 2, 3]$, $right = [4, 5]$. We choose to keep the $left$ with the sum equalling 6.

Explanation of the second test case:

1. It can be demonstrated that an array with the sum $s_1 = 1$ is imposssible to generate. We can get an array with the sum $s_2 = 2$ in the following way:
   a. $a = [3, 1, 3, 1, 3]$, $mid = \frac{1+3}{2} = 2$, $left = [1, 1]$, $right = [3, 3, 3]$. We choose to keep the left array with the sum equalling 2.
2. It can be demonstrated that an array with the sum $s_3 = 3$ is imposssible to generate. We can get an array with the sum $s_4 = 9$ in the following way:
   a. $a = [3, 1, 3, 1, 3]$, $mid = \frac{1+3}{2} = 2$, $left = [1, 1]$, $right = [3, 3, 3]$. We choose to keep the $right$ array with the sum equalling 9.
   b. We can get an array with the sum $s_5 = 11$ with zero slicing operations, because array sum is equal to 11.