

Practice Set

Total no of questions: 12

1. Given a dictionary in Python, write a program to find the highest 3 values in a dictionary.

Input : `my_dict = {'A': 67, 'B': 23, 'C': 45,
 'D': 56, 'E': 12, 'F': 69}`

Output : `{'F': 69, 'A': 67, 'D': 56}`

Reference Link:

<https://www.geeksforgeeks.org/python-program-to-find-the-highest-3-values-in-a-dictionary/>

2. Given a list of dictionaries in Python, write a Python program to convert List of Dictionaries to List of Lists.

Input1 : `test_list = [{'Gfg': 123, 'best': 10}, {'Gfg': 51, 'best': 7}]`

Output 1: `[['Gfg', 'best'], [123, 10], [51, 7]]`

Input 2: `test_list = [{'Gfg': 12}]`

Output2 : `[['Gfg'], [12]]`

Reference Link:

<https://www.geeksforgeeks.org/python-convert-list-of-dictionaries-to-list-of-lists/?ref=leftbar-rightbar>

3. Chef has a sequence A_1, A_2, \dots, A_N . This sequence has exactly 2^N subsequences. Chef considers a subsequence of A interesting if its size is exactly K and the sum of all its elements is minimum possible, i.e. there is no subsequence with size K which has a smaller sum.

Help Chef find the number of interesting subsequences of the sequence A .

Input

The first line of the input contains a single integer T denoting the number of test cases. The description of T test cases follows.

The first line of each test case contains two space-separated integers N and K .

The second line contains N space-separated integers A_1, A_2, \dots, A_N .

Output

For each test case, print a single line containing one integer — the number of interesting subsequences.

Constraints

$$1 \leq T \leq 10$$

$$1 \leq K \leq N \leq 50$$

$$1 \leq A_i \leq 100 \text{ for each valid } i$$

Subtasks

Subtask #1 (30 points): $1 \leq N \leq 20$

Subtask #2 (70 points): original constraints

Sample Input 1

1

4 2

1 2 3 4

Sample Output 1

1

Explanation

Example case 1: There are six subsequences with length 2: (1,2), (1,3), (1,4), (2,3), (2,4) and (3,4). The minimum sum is 3 and the only subsequence with this sum is (1,2).

Reference Link : <https://www.codechef.com/problems/CHEFINSQ>

4. Appy and Chef are participating in a contest. There are N problems in this contest; each problem has a unique problem code between 1 and N inclusive. Appy and Chef decided to split the problems to solve between them — Appy should solve the problems whose problem codes are divisible by A but not divisible by B , and Chef should solve the problems whose problem codes are divisible by B but not divisible by A (they decided to not solve the problems whose codes are divisible by both A and B).

To win, it is necessary to solve at least K problems. You have to tell Appy whether they are going to win or lose.

Input

The first line of the input contains a single integer T denoting the number of test cases. The description of T test cases follows.

The first and only line of each test case contains four space-separated integers N , A , B and K .

Output

For each test case, print a single line containing the string "Win" if they can solve at least K problems or "Lose" otherwise (without quotes).

Constraints

$$1 \leq T \leq 15$$

$$1 \leq K \leq N \leq 10^8$$

$$1 \leq A, B \leq 10^9$$

Subtasks

Subtask #1 (15 points):

$$1 \leq T \leq 15$$

$$1 \leq K \leq N \leq 10^6$$

$$1 \leq A, B \leq 10^3$$

Subtask #2 (85 points): original constraints

Input :

1

6 2 3 3

Output :

Win

Explanation

Example case 1: Appy is solving the problems with codes 2 and 4, Chef is solving the problem with code 3. Nobody is solving problem 6, since 6 is divisible by both 2 and 3. Therefore, they can solve 3 problems and win.

Reference Link : <https://www.codechef.com/problems/HMAPPY2>

Q5. Jeff's got n cards, each card contains either digit 0, or digit 5. Jeff can choose several cards and put them in a line so that he gets some number. What is the largest possible number divisible by 90 Jeff can make from the cards he's got?

Jeff must make the number without leading zero. At that, we assume that number 0 doesn't contain any leading zeros. Jeff doesn't have to use all the cards.

Input

The first line contains integer n ($1 \leq n \leq 103$). The next line contains n integers a_1, a_2, \dots, a_n ($a_i = 0$ or $a_i = 5$). Number a_i represents the digit that is written on the i -th card.

Output

In a single line print the answer to the problem — the maximum number, divisible by 90. If you can't make any divisible by 90 number from the cards, print -1.

Examples

Input

4
5 0 5 0

Output

0

Input

11
5 5 5 5 5 5 5 0 5 5

Output

5555555550

Reference Link: <https://codeforces.com/problemset/problem/352/A>

Q.6 You are given a string s consisting of n lowercase Latin letters. n is even.

For each position i ($1 \leq i \leq n$) in string s you are required to change the letter on this position either to the previous letter in alphabetic order or to the next one (letters 'a' and 'z' have only one of these options). Letters in every position must be changed exactly once.

For example, letter 'p' should be changed either to 'o' or to 'q', letter 'a' should be changed to 'b' and letter 'z' should be changed to 'y'.

That way string "codeforces", for example, can be changed to "dpedepqbft" ('c' \rightarrow 'd', 'o' \rightarrow 'p', 'd' \rightarrow 'e', 'e' \rightarrow 'd', 'f' \rightarrow 'e', 'o' \rightarrow 'p', 'r' \rightarrow 'q', 'c' \rightarrow 'b', 'e' \rightarrow 'f', 's' \rightarrow 't').

String s is called a palindrome if it reads the same from left to right and from right to left. For example, strings "abba" and "zz" are palindromes and strings "abca" and "zy" are not.

Your goal is to check if it's possible to make string s a palindrome by applying the aforementioned changes to every position. Print "YES" if string s can be transformed to a palindrome and "NO" otherwise.

Each testcase contains several strings, for each of them you are required to solve the problem separately.

Input

The first line contains a single integer T ($1 \leq T \leq 50$) — the number of strings in a testcase.

Then $2T$ lines follow — lines $(2i-1)$ and $2i$ of them describe the i -th string. The first line of the pair contains a single integer n ($2 \leq n \leq 100$, n is even) — the length of the corresponding string. The second line of the pair contains a string s , consisting of n lowercase Latin letters.

Output

Print T lines. The i -th line should contain the answer to the i -th string of the input. Print "YES" if it's possible to make the i -th string a palindrome by applying the aforementioned changes to every position. Print "NO" otherwise.

Input

5

6

abccba

2

cf

4

adfa

8

abaazaba

2

ml

Output

YES

NO

YES

NO

NO

Explanation

The first string of the example can be changed to "bcbcb", two leftmost letters and two rightmost letters got changed to the next letters, two middle letters got changed to the previous letters.

The second string can be changed to "be", "bg", "de", "dg", but none of these resulting strings are palindromes.

The third string can be changed to "beeb" which is a palindrome.

The fifth string can be changed to "lk", "lm", "nk", "nm", but none of these resulting strings are palindromes. Also note that no letter can remain the same, so you can't obtain strings "ll" or "mm".

Reference Link: <https://codeforces.com/problemset/problem/1027/A>

Q7. Majority element- Given an array `nums` of size `n`, return *the majority element*.

The majority element is the element that appears more than $\lfloor n / 2 \rfloor$ times. You may assume that the majority element always exists in the array.

Example 1:

Input: `nums = [3,2,3]`

Output: 3

Example 2:

Input: `nums = [2,2,1,1,1,2,2]`

Output: 2

Constraints:

- `n == nums.length`
- `1 <= n <= 5 * 104`

- $-231 \leq \text{nums}[i] \leq 231 - 1$

Follow-up: Could you solve the problem in linear time and in $O(1)$ space?

Reference Link: <https://leetcode.com/problems/majority-element/>

Q8 Factorial of a very large number

Given an integer N, find its factorial.

Example 1:

Input: N = 5

Output: 120

Explanation : $5! = 1*2*3*4*5 = 120$

Example 2:

Input: N = 10

Output: 3628800

Explanation :

$10! = 1*2*3*4*5*6*7*8*9*10 = 3628800$

Your Task:

You don't need to read input or print anything. Complete the function *factorial()* that takes integer N as input parameter and returns a list of integers denoting the digits that make up the factorial of N.

Expected Time Complexity : $O(N^2)$

Expected Auxilliary Space : $O(1)$

Constraints:

$1 \leq N \leq 1000$;

Reference

Link:-<https://practice.geeksforgeeks.org/problems/factorials-of-large-numbers2508/1>

Q9. Maximize sum after K negations

Given an array of integers of size N and a number K., You must modify array arr[] exactly K number of times. Modifying an array means in each operation you can replace any array element either arr[i] by -arr[i] or -arr[i] by arr[i]. You need to perform this operation in such a way that after K operations, the sum of the array must be maximum.

Example 1:

Input:

N = 5, K = 1

arr[] = {1, 2, -3, 4, 5}

Output:

15

Explanation:

We have k=1 so we can change -3 to 3 and sum all the elements to produce 15 as output.

Example 2:

Input:

N = 10, K = 5

arr[] = {5, -2, 5, -4, 5, -12, 5, 5, 5, 20}

Output:

68

Explanation:

Here we have $k=5$ so we turn -2, -4, -12 to 2, 4, and 12 respectively. Since we have performed 3 operations so k is now 2. To get the maximum sum of the array we can turn positive turned 2 into negative and then positive again so k is 0. Now sum is $5+5+4+5+12+5+5+5+20+2 = 68$

Your Task:

You **don't** have to print anything, printing is done by the driver code itself. You have to complete the function **maximizeSum()** which takes the array **A[]**, its size **N**, and an integer **K** as inputs and returns the maximum possible sum.

Expected Time Complexity: $O(N \cdot \log N)$

Expected Auxiliary Space: $O(1)$

Constraints:

$$1 \leq N, K \leq 10^5$$

$$-10^9 \leq A_i \leq 10^9$$

Reference Link:

[https://practice.geeksforgeeks.org/problems/maximize-sum-after-k-negations1149/1/?page=4&difficulty\[\]=0&category\[\]=Arrays&sortBy=submissions](https://practice.geeksforgeeks.org/problems/maximize-sum-after-k-negations1149/1/?page=4&difficulty[]=0&category[]=Arrays&sortBy=submissions)

Q10. Minimum number of flips

Given a binary string, that is it contains only 0s and 1s. We need to make this string a sequence of alternate characters by flipping some of the bits, our goal is to minimize the number of bits to be flipped.

Example 1:**Input:**

$S = "001"$

Output: 1

Explanation:

We can flip the 0th bit to 1 to have
101.

Example 2:**Input:**

S = "0001010111"

Output: 2

Explanation: We can flip the 1st and 8th bit
to have "0101010101"
101.

Your Task:

You don't need to read input or print anything. Your task is to complete the function **minFlips()** which takes the string S as input and returns the minimum number of flips required.

Expected Time Complexity: $O(|S|)$.

Expected Auxiliary Space: $O(1)$.

Constraints:

$1 \leq |S| \leq 10^5$

Reference Link:

[https://practice.geeksforgeeks.org/problems/min-number-of-flips3210/1/?page=2&difficulty\[\]=0&category\[\]=Strings&sortBy=submissions](https://practice.geeksforgeeks.org/problems/min-number-of-flips3210/1/?page=2&difficulty[]=0&category[]=Strings&sortBy=submissions)

Q11. Richest Customer Wealth

You are given an $m \times n$ integer grid **accounts** where **accounts[i][j]** is the amount of money the *i*th customer has in the *j*th bank. Return the wealth that the richest customer has.

A customer's wealth is the amount of money they have in all their bank accounts. The richest customer is the customer that has the maximum wealth.

Example 1:

Input: accounts = [[1,2,3],[3,2,1]]

Output: 6

Explanation:

1st customer has wealth = $1 + 2 + 3 = 6$

2nd customer has wealth = $3 + 2 + 1 = 6$

Both customers are considered the richest with a wealth of 6 each, so return 6.

Example 2:

Input: accounts = [[1,5],[7,3],[3,5]]

Output: 10

Explanation:

1st customer has wealth = 6

2nd customer has wealth = 10

3rd customer has wealth = 8

The 2nd customer is the richest with a wealth of 10.

Example 3:

Input: accounts = [[2,8,7],[7,1,3],[1,9,5]]

Output: 17

Reference Link: <https://leetcode.com/problems/richest-customer-wealth/>

Q12 Find All Possible Recipes from Given Supplies

You have information about n different recipes. You are given a string array `recipes` and a 2D string array `ingredients`. The i th recipe has the name `recipes[i]`, and you can create it if you have all the needed ingredients from `ingredients[i]`. Ingredients to a recipe may need to be created from other recipes, i.e., `ingredients[i]` may contain a string that is in `recipes`.

You are also given a string array `supplies` containing all the ingredients that you initially have, and you have an infinite supply of all of them.

Return *a list of all the recipes that you can create*. You may return the answer in any order.

Note that two recipes may contain each other in their ingredients.

Example 1:

Input: `recipes = ["bread"], ingredients = [["yeast","flour"]], supplies = ["yeast","flour","corn"]`

Output: `["bread"]`

Explanation:

We can create "bread" since we have the ingredients "yeast" and "flour".

Example 2:

Input: `recipes = ["bread","sandwich"], ingredients = [["yeast","flour"],["bread","meat"]], supplies = ["yeast","flour","meat"]`

Output: `["bread","sandwich"]`

Explanation:

We can create "bread" since we have the ingredients "yeast" and "flour".

We can create "sandwich" since we have the ingredient "meat" and can create the ingredient "bread".

Example 3:

Input: recipes = ["bread","sandwich","burger"], ingredients =
[["yeast","flour"],["bread","meat"],["sandwich","meat","bread"]], supplies =
["yeast","flour","meat"]

Output: ["bread","sandwich","burger"]

Explanation:

We can create "bread" since we have the ingredients "yeast" and "flour".

We can create "sandwich" since we have the ingredient "meat" and can create the ingredient "bread".

We can create "burger" since we have the ingredient "meat" and can create the ingredients "bread" and "sandwich".

Reference Link:

<https://leetcode.com/problems/find-all-possible-recipes-from-given-supplies/>