unacademy					

Words that contain many consecutive consonants, like "schtschurowskia", are generally considered somewhat hard to pronounce.

We say that a word is *hard to pronounce* if it contains **4** or more consonants in a row; otherwise it is *easy to pronounce*. For example, "apple" and "polish" are easy to pronounce, but "schtschurowskia" is hard to pronounce.

You are given a string S consisting of N lowercase Latin characters. Determine whether it is easy to pronounce or not based on the rule above — print YES if it is easy to pronounce and NO otherwise.

For the purposes of this problem, the vowels are the characters  $\{a, e, i, o, u\}$  and the consonants are the other 21 characters.

## **Input Format**

- ullet 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.
  - $\circ$  The first line of each test case contains a single integer N, the length of string S.
  - $\circ$  The second line of each test case contains the string S.

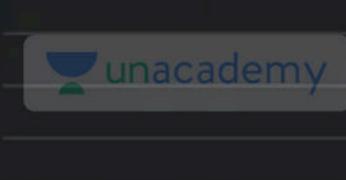
## **Output Format**

For each test case, output on a new line the answer — YES if S is easy to pronounce, and NO otherwise.

Each character of the output may be printed in either uppercase or lowercase. For example, the strings YES, yeS, yes, and YeS will all be treated as identical.

#### **Constraints**

- $1 \le T \le 100$
- $1 \le N \le 100$
- S contains only lowercase Latin characters, i.e. the characters  $\{a,b,c,\ldots,z\}$



Chef is trying to solve a problem having T test cases, where, for each test case he is given a single integer N.

Chef has an algorithm which takes **exactly**  $N^2$  iterations for a test case with value N.

The constraints of the problem are as follows:

- $1 \le T \le maxT$
- $1 \le N \le maxN$
- Sum of N over all test cases does not exceed sumN.

Given the values maxT, maxN, and sumN, determine the **maximum** number of iterations Chef's algorithm can take for any valid input file satisfying all the constraints.

Formally speaking, find the **maximum** value of  $N_1^2 + N_2^2 + \cdots + N_T^2$  for:

- $1 \le T \le maxT$
- $1 \leq N_i \leq maxN$
- $N_1 + N_2 + N_3 + \cdots + N_T \leq sumN$

## **Input Format**

- ullet The first line of input will contain a single integer T, denoting the number of test cases.
- Each test case consists of single line consisting of three integers maxT, maxN, and sumN.

## **Output Format**

For each test case, output the the **maximum** number of iterations Chef's algorithm can take in any valid test file.

### Constraints

- $1 \le T \le 1000$
- $\sim 1 < man T < man N < aum N < 104$



### **Problem**

CodeChef offers a feature called *streak count*. A streak is maintained if you solve **at least one** proble daily.

Om and Addy actively maintain their streaks on CodeChef. Over a span of N consecutive days, you have observed the count of problems solved by each of them.

Your task is to determine the **maximum** streak achieved by Om and Addy and find who had the lon; maximum streak.

#### Input Format

- ullet The first line of input will contain a single integer T, denoting the number of test cases.
- Each test case consists of multiple lines of input.
- $\circ$  The first line of each test case contains an integer N the number of days.
- $\circ$  The second line of each test case contains N space-separated integers, the  $i^{th}$  of which is  $A_i$ , representing the problems solved by Om on the  $i^{th}$  day.
- $\circ$  The third line of each test case contains N space-separated integers, the  $i^{th}$  of which is  $B_i$ , representing the problems solved by Addy on the  $i^{th}$  day.

#### **Output Format**

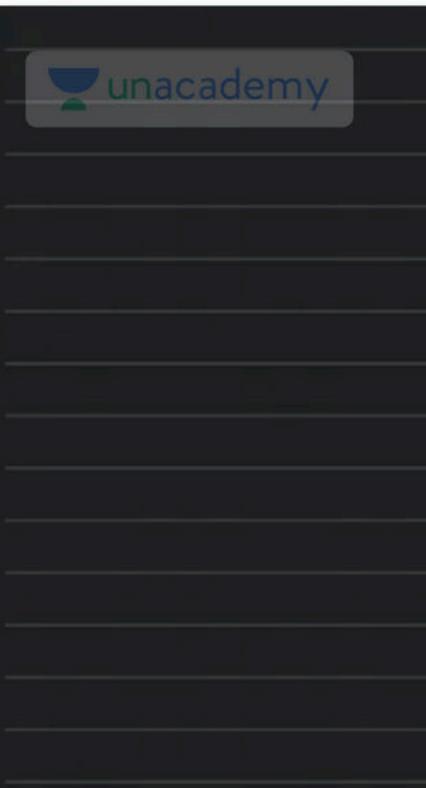
For each test case, output:

- 0M, if Om has longer maximum streak than Addy;
- ADDY, if Addy has longer maximum streak than Om;
- DRAW, if both have equal maximum streak.

You may print each character in uppercase or lowercase. For example, OM, om, Om, and oM, are all considered the same.

#### Constraints

- $1 \le T \le 10^5$
- $1 \le N \le 10^5$
- $0 \le A_i, B_i \le 10^9$
- . The sum of M ever all test asses wen't evered 6 10



# **Problem**

Kulyash believes in equality.

Given an integer N, output a binary string of length N such that:

- $\bullet$  The count of 01 subsequences in the string is **equal** to the count of 10 subsequences;
- The string has **at least** one occurrence of 0 as well as 1.

If multiple such strings exist, print any. Also, it is guaranteed that corresponding to the given input, an answer always exists.

### **Input Format**

- ullet First line will contain T, number of test cases. Then the test cases follow.
- ullet Each test case contains of a single line of input, an integer N the length of the binary string.

## **Output Format**

For each test case, output any binary string of length N satisfying the given conditions.

#### **Constraints**

- $1 \leq T \leq 100$
- $3 \le N \le 1000$

### **Subtasks**

## Sample 1:

Input	Ō	Output	Ō
2		1001	
3		010	