

А. Удалить наименьшее

1 секунда, 256 мегабайт

Вам дан массив a , состоящий из n положительных (больших нуля) целых чисел.

За один ход вы можете выбрать два индекса i и j ($i \neq j$) таких, что абсолютная разность между a_i и a_j не превосходит единицу ($|a_i - a_j| \leq 1$), и удалить наименьший из этих двух элементов. Если два элемента равны, вы можете удалить любой из них (но только один).

Ваша задача — определить, возможно ли получить массив, состоящий **только из одного элемента**, применив несколько (возможно, ноль) таких ходов, или нет.

Вам нужно ответить на t независимых наборов тестовых данных.

Входные данные

Первая строка теста содержит одно целое число t ($1 \leq t \leq 1000$) — количество наборов тестовых данных. Затем следуют t наборов тестовых данных.

Первая строка набора тестовых данных содержит одно целое число n ($1 \leq n \leq 50$) — длину a . Вторая строка набора тестовых данных содержит n целых чисел a_1, a_2, \dots, a_n ($1 \leq a_i \leq 100$), где a_i — i -й элемент a .

Выходные данные

Для каждого набора тестовых данных выведите ответ на него: «YES», если возможно получить массив, состоящий **только из одного элемента**, применив несколько (возможно, ноль) ходов, описанных в условии задачи, или «NO» в обратном случае.

входные данные
5 3 1 2 2 4 5 5 5 5 3 1 2 4 4 1 3 4 4 1 100
выходные данные
YES YES NO NO YES

В первом наборе тестовых данных примера мы можем применить следующую последовательность ходов:

- выбрать $i = 1$ и $j = 3$, а затем удалить a_i (таким образом, a превратится в $[2; 2]$);
- выбрать $i = 1$ и $j = 2$, а затем удалить a_j (таким образом, a превратится в $[2]$).

Во втором наборе тестовых данных примера мы можем выбирать любые возможные i и j в течение любого хода и удалять любой элемент из пары.

В третьем наборе тестовых данных примера не существует способа избавиться от 2 и 4.

В. Карточки

2 секунды, 256 мегабайт

Когда Серёже было три года, ему подарили на день рождения набор карточек с буквами. С их помощью было записано словамилюбимое число мамы мальчика в двоичной системе счисления. Серёжа тотчас же принялся с ними играть, но так как не умел читать, перемешал их в случайном порядке. Папа решил привести в порядок карточки. Помогите ему восстановить исходное число при условии, что оно было максимально возможным.

Входные данные

Первая строка содержит одно целое число n ($1 \leq n \leq 10^5$) — длина строки. Вторая строка содержит строку из строчных латинских букв: «z», «e», «r», «o» и «n».

Гарантируется, что буквы можно переупорядочить таким образом, чтобы они образовывали последовательность слов, каждое из которых является либо словом «zero», что соответствует цифре 0, либо словом «one», что соответствует цифре 1.

Выходные данные

Выведите максимально возможное число в двоичной системе счисления. Выводите двоичные цифры, разделяя их пробелами. Лидирующие нули допустимы.

входные данные
4 ezor
выходные данные
0

входные данные
10 nznnooeoer
выходные данные
1 1 0

В первом примере правильная последовательность букв — это «zero».

Во втором примере правильная последовательность букв — это «oneonezero».

С. Формирование команд

1 секунда, 256 мегабайт

В университете учатся n студентов. Количество студентов чётно. Умение i -го студента программировать равно a_i .

Тренер хочет сформировать $\frac{n}{2}$ команд. Каждая команда должна состоять ровно из двух студентов, а каждый студент должен принадлежать ровно одной команде. Два студента могут сформировать команду только тогда, когда их умение программировать одинаково (иначе они не смогут понять друг друга и не смогут сформировать команду).

Студенты могут решать задачи, чтобы улучшать их умение программировать. Одна решенная задача увеличивает умение программировать на один.

Тренер хочет знать, какое минимальное количество задач студентам необходимо решить, чтобы сформировать ровно $\frac{n}{2}$ команд (то есть каждая пара студентов должна формировать команду). Ваша задача — найти это количество.

Входные данные

Первая строка входных данных содержит одно целое число n ($2 \leq n \leq 100$) — количество студентов. Гарантируется, что n всегда является чётным числом.