

Задача A (“RLE”)

Будем рассматривать только строчки, состоящие из заглавных латинских букв. Например, рассмотрим строку AAAABCCCCCDDDD. Длина этой строки равна 14. Поскольку строка состоит только из латинских букв, повторяющиеся символы могут быть удалены и заменены числами, определяющими количество повторений. Таким образом, данная строка может быть представлена как 4A5C4D. Длина такой строки 7. Описанный метод мы назовем упаковкой строки.

Напишите программу, которая берет упакованную строчку и восстанавливает по ней исходную строку.

Формат входных данных

На вход подаётся одна упакованная строка. В строке могут встречаться только конструкции вида nA , где n — количество повторений символа (целое число от 2 до 99), а A — заглавная латинская буква, либо конструкции вида A , то есть символ без числа, определяющего количество повторений.

Максимальная длина строки не превышает 80.

Формат выходных данных

В выходной файл выведите восстановленную строку. При этом строка должна быть разбита на строчки длиной ровно по 40 символов (за исключением последней, которая может содержать меньше 40 символов).

Примеры

Вход

3A4B7D

Выход

AAABBBDDDDDDDD

Вход

22D7AC18FGD

Выход

DDDDDDDDDDDDDDDDDDDDDDDDAAAAAACFFFFFFFFF
FFFFFFFFFGD

Задача В (“Видеокарты или интеграл Лебега”)

Маленький Влад обожает играть в популярную игру Vota-2 на своем компьютере. На днях разработчики сообщили о выходе её продолжения под названием Vota-3. Разумеется, Влад тут же её приобрёл, но оказалось, что компьютер Влада слишком старый для этой игры, поэтому он отправился в магазин за обновлением для него.

В магазине Влад увидел n видеокарт, i -я из которых имеет свою мощность a_i , выражаемую целым положительным числом. Для того, чтобы игра точно заработала, Влад решил купить не одну, а несколько видеокарт и объединить их мощности по новейшей технологии. Для использования этой технологии одна из видеокарт выбирается в качестве ведущей, а все остальные видеокарты подключаются к ней в качестве вторичных. Для корректной работы этой конструкции необходимо, чтобы мощность каждой видеокарты была кратна мощности ведущей видеокарты. Для обеспечения совместимости мощность каждой вторичной видеокарты может быть понижена до любого меньшего целого положительного значения. При этом мощность опорной видеокарты должна оставаться оригинальной, то есть не может быть уменьшена.

У Влада очень много денег, поэтому он может купить любые видеокарты. Помогите ему выбрать такие видеокарты, чтобы после необходимого понижения их мощностей и соединения их по новой технологии суммарная итоговая мощность видеокарт была максимальной.

Входные данные

В первой строке находится целое число n ($1 \leq n \leq 200\,000$) — количество видеокарт в магазине.

Во второй строке находится n целых чисел a_1, a_2, \dots, a_n ($1 \leq a_i \leq 200\,000$), обозначающих мощности видеокарт.

Выходные данные

В единственной строке выведите целое число — максимальную возможную суммарную итоговую мощность видеокарт.

Примеры

Входные данные

```
4
3 2 15 9
```

Выходные данные

```
27
```

Входные данные

```
4
8 2 2 7
```

Выходные данные

```
18
```

Задача С (“Дремучий лес”)

Пресека — это такая прямая линия, которая проходит через лес (то есть деревья есть как с одной стороны от этой линии, так и с другой), и при этом она не проходит ни через одно из деревьев леса, а также не касается деревьев. Будем говорить, что лес является дремучим, если в нем нет ни одной пресеки.

На плане леса все деревья изображаются кругами. Никакие два круга не пересекаются и не касаются друг друга. Требуется по этому плану определить, является ли лес дремучим.

Формат входных данных

Сначала на вход подаётся целое число N — количество деревьев ($1 \leq N \leq 200$). Затем идут N троек чисел, задающих деревья. Первые два числа задают координаты центра, а третье — радиус. Все данные задаются точно, и выражаются вещественными числами, не более чем с 2 знаками после десятичной точки, по модулю не превосходящими 1000.

Формат выходных данных

В первой строке выходного файла должно содержаться сообщение YES, если лес является дремучим, и NO иначе. Во втором случае вторая строка выходного файла должна содержать координаты двух точек, через которые проходит пресека. Все координаты не должны превышать 2000, и расстояние между выданными точками должно быть не меньше 100.

Примеры

Вход

```
3
0 10 2
5 11 2
12.04 7 2
```

Выход

```
NO
2.5 0
2.5 100
```

Вход

```
3
0 0 1
2.05 0 1
1.02 -1.9 1
```

Выход

```
YES
```

Задача D (“Перегоны”)

На некоторой железнодорожной ветке расположено N станций, которые последовательно пронумерованы числами от 1 до N . Известны расстояния между некоторыми станциями. Требуется точно вычислить длины всех перегонов между соседними станциями или указать, что это сделать невозможно (то есть приведенная информация является противоречивой или ее недостаточно).

Формат входных данных

Во входном файле записаны сначала числа N — количество станций ($2 \leq N \leq 100$) и E — количество пар станций, расстояния между которыми заданы ($0 \leq E \leq 10000$). Далее идет E троек чисел, первые два числа каждой тройки задают номера станций (это числа из диапазона от 1 до N), а третье — расстояние между этими станциями (все эти расстояния заданы точно и выражаются вещественными неотрицательными числами не более чем с 3-я знаками после десятичной точки).

Формат выходных данных

В случае, когда восстановить длины перегонов можно однозначно, в выходной файл выведите сначала число 1, а затем $N-1$ вещественное число. Первое из этих чисел должно соответствовать расстоянию от 1-й станции до 2-й, второе — от 2-й до 3-й, и так далее. Все числа должны быть выведены с точностью до 3-х знаков после десятичной точки.

Если приведенная информация о расстояниях между станциями является противоречивой или не позволяет однозначно точно восстановить длины перегонов, выведите в выходной файл одно число 2.

Примеры

Вход

```
3 2
1 2 1.250
3 1 3
```

Выход

```
1
1.250 1.750
```

Вход

```
3 1
1 1 1
```

Выход

```
2
```