

"А" Универсальная розетка
Ограничение по времени 2 секунды
Ограничение по памяти 256 мегабайт

Электрическая розетка устроена следующим образом: у нее есть два контакта, к каждому из которых подведено по проводу. Эти контакты (как и подведенные к ним провода) принято называть, соответственно, «фаза» и «ноль». На внешней панели электрической розетки есть два отверстия, которые служат для подключения к ней электрических приборов.

Электрические приборы подключаются к розетке при помощи так называемых вилок. Вилка имеет два контакта, которые вставляются в отверстия розетки. При включении вилки в розетку соответствующие контакты вилки и розетки замыкаются.

В разных странах мира используются вилки с разным расстоянием между контактами, что создает серьезные проблемы для путешественников. Одна крупная компания решила выпустить уникальный продукт: розетку, в которую можно будет воткнуть вилку практически любой страны! Эта розетка имеет n отверстий. К части из них подведен ноль, а к остальным подведена фаза.

Ваша задача — определить, можно ли подключить вилку с расстоянием между контактами d к данной розетке таким образом, чтобы один контакт вилки был замкнут на ноль, а второй — на фазу.

Формат входных данных

Первая строка содержит два целых числа n и d ($1 \leq n \leq 200$, $1 \leq d \leq 10000$).

Введем на внешней панели розетки прямоугольную декартову систему координат. В следующих n строках содержатся по три целых числа x_i , y_i , t_i ($-10000 \leq x_i, y_i \leq 10000$; $0 \leq t_i \leq 1$). Эта запись обозначает, что в точке с координатами x_i , y_i находится отверстие. Если $t_i = 1$, то этому отверстию соответствует контакт, к которому подведена фаза. Если $t_i = 0$, то этому отверстию соответствует контакт, к которому подведен ноль.

Все отверстия находятся в различных точках.

Формат выходных данных

Выведите Yes, если данную вилку можно корректно подключить к данной розетке, и No в противном случае.

Входные данные

2 1
0 0 0
0 1 1

Выходные данные

Yes

Входные данные

2 2
0 0 0
0 1 1

Выходные данные

No

"В" Граница

Ограничение по времени 2 секунды

Ограничение по памяти 256 мегабайт

В международной политике важным понятием является граница между государствами. Нечеткое понимание сторонами того, где проходит граница, может привести к международным конфликтам и даже войнам.

В этой задаче ситуация обстоит несколько проще, так как у двух рассматриваемых в задаче государств есть четкое понимание, какая территория принадлежит какому из них. Территория, занимаемая этими двумя государствами, представляет собой прямоугольник размером h на w километров, разбитый на квадраты со стороной в один километр. Каждый из этих квадратов полностью принадлежит либо первому государству, либо второму.

Необходимо определить длину границы между двумя государствами. Сторона единичного квадрата считается принадлежащей границе, если по одну сторону от нее лежит квадрат, принадлежащий первому государству, а по другую — принадлежащий второму.

Формат входных данных

Первая строка содержит два целых числа: w и h — размеры прямоугольника в километрах ($1 \leq w, h \leq 100$). Далее следуют h строк, описывающих территорию. Каждая из них содержит w символов. Если символ равен А, то соответствующий единичный квадрат принадлежит первому государству, а если он равен В, то второму. Гарантируется, что каждому государству принадлежит хотя бы один квадрат.

Территории каждого из государств представляют собой связные области.

Формат выходных данных

Выведите одно целое число — длину границы между государствами в километрах.

Входные данные

5 6

AAABV

ABBBV

AAABV

AAAAB

AAAAB

AABBB

Выходные данные

13

"С" Загранпаспорт

Ограничение по времени 2 секунды

Ограничение по памяти 256 мегабайт

Многим знакома ситуация, когда для получения какого-либо документа, например, загранпаспорта, требуется в строго определенном порядке обойти несколько мест, в каждом из которых требуется совершить какое-либо действие: получить справку, написать заявление, заверить ксерокопию и т.д. Для удобства граждан такие места стали объединяться в единые центры, где каждое место превращается в отдельное окно. Но все равно есть проблема: каждое окно имеет свой режим работы.

Один человек планирует прийти за своим загранпаспортом в hh часов mm минут. Он знает, что ему требуется обойти в определенном порядке ровно n окон, и знает их режим работы. В конце обслуживания в последнем из них он получит загранпаспорт. Человек хочет узнать, успеет ли он получить паспорт до конца дня, если кроме него в едином центре не будет других посетителей.

Про каждое окно известно его время открытия и время закрытия. Также известно, сколько минут занимает обслуживание одного посетителя в каждом окне.

Считается, что посетитель подходит к окну в начале некоторой минуты. Время открытия окна — это первая минута, во время которой оно уже работает, а время закрытия — первая минута, когда оно уже не работает. Например, если окно открывается в 12:00 и закрывается в 20:00, а обслуживание занимает 11 минут, то если человек подошел к окну между 12:00 и 19:49, включительно, то его обслужат сразу, если в 11:59 или раньше, то его начнут обслуживать в 12:00, а если в 19:50 или позже, то его уже не обслужат.

Человек перемещается между окнами мгновенно. Таким образом, например, если обслуживание в некотором окне занимает 10 минут, и человек подошел к нему в 12:45, то обслуживание в следующем окне для него может начаться в 12:55 или позже.

Все окна открываются не раньше, чем в 00:00, и закрываются не позже, чем в 23:00. В окне не обслуживают посетителя, если до конца рабочего времени окна осталось меньше времени, чем требуется на обслуживание.

Требуется выяснить, успеет ли человек получить загранпаспорт, и если да, то в какой самый ранний момент времени он сможет покинуть единый центр, получив загранпаспорт.

Формат входных данных

Первая строка содержит время, когда человек планирует прийти в единый центр, в формате $hh:mm$. Вторая строка содержит одно целое число n — количество окон, которые требуется обойти ($1 \leq n \leq 100$).

Следующие n строк содержат описание работы всех окон в том порядке, в котором их нужно обходить. Строка, описывающая окно номер i , содержит время открытия этого окна в формате $hh:mm$, затем через пробел время закрытия в том же формате, и через пробел одно целое число ti — количество минут, которые займет обслуживание посетителя в этом окне ($1 \leq ti \leq 1440$). Время закрытия каждого окошка строго больше времени его открытия.

Формат выходных данных

В случае, если человек успеет получить загранпаспорт, выведите в первой строке Yes, а во второй — время, когда он сможет покинуть единый центр в формате $hh:mm$. Иначе выведите в первой строке No.

Входные данные

08:00

2

09:05 11:05 30

09:00 10:20 45

Выходные данные

Yes

10:20

Входные данные

08:00

2

10:00 12:00 35

10:00 11:15 45

Выходные данные

No

"D" Почтовое отправление

Ограничение по времени 2 секунды

Ограничение по памяти 256 мегабайт

Для подготовки заключительного этапа Russian Code Cup организаторам потребовалось отправить на место проведения n предметов. Для каждого предмета известна его масса в граммах m_i .

Для отправки решено было воспользоваться почтовой службой «Суперэкспресс». Служба принимает к пересылке бандероли, в каждой из которых может пересылаться один или несколько предметов. При этом масса бандероли равна сумме масс пересылаемых в ней предметов.

Пересылка бандероли стоит 1 рубль за грамм, за исключением бандеролей, которые попадают под действие специального предложения. А именно, если бандероль весит ровно один килограмм, то стоимость ее пересылки составляет P рублей.

Организаторы Russian Code Cup хотят переслать все предметы, затратив минимальную возможную сумму денег. Помогите им распределить предметы по бандеролям, чтобы добиться этого.

Формат входных данных

Первая строка содержит два целых числа: n и P ($1 \leq n \leq 14$; $1 \leq P \leq 1000$) — количество предметов и стоимость пересылки бандероли в рамках специального предложения.

Вторая строка содержит n целых чисел: $m_1, m_2, \dots, m_n$ ($1 \leq m_i \leq 1000$ для всех i от 1 до n).

Формат выходных данных

Выведите одно число — минимальную суммарную стоимость пересылки всех предметов в рублях.

Входные данные

5 800

100 200 300 400 500

Выходные данные

1300

Входные данные

5 800

400 400 400 400 400

Выходные данные

2000

"Е" Сортировка очередями

Ограничение по времени 2 секунды

Ограничение по памяти 256 мегабайт

Очередь — структура данных, основанная на принципе «первый пришел — первый вышел» (FIFO, First In — First Out). Добавление элемента возможно только в конец очереди, извлечение — только из начала очереди. Таким образом, элементы извлекаются из очереди в том же порядке, в котором они были в нее добавлены.

Задача сортировки состоит в упорядочивании заданного массива чисел (или других объектов) по возрастанию или убыванию. У этой задачи существует достаточно многих вариантов, для многих из которых существуют весьма эффективные алгоритмы.

Далее в задаче рассматривается специальное устройство, содержащее входной поток, выходной поток и k очередей, пронумерованных числами от 1 до k .

Ваша задача состоит в том, чтобы по заданной последовательности различных чисел, которую требуется отсортировать, составить программу для этого устройства, которая будет выводить в выходной поток те же числа, что поступили во входной поток, но упорядоченные по возрастанию.

Программа должна содержать ровно $2n$ операций, каждая из которых либо читает число из входного потока и добавляет его в одну из очередей, либо извлекает число из одной из очередей и выводит его в выходной поток.

Формат входных данных

Первая строка содержит целое число n ($1 \leq n \leq 300$). Вторая строка содержит n различных целых чисел $a_1, \dots, a_n$ в том порядке, в котором они поступают из входного потока ($1 \leq a_i \leq 10^9$ для всех i от 1 до n). Третья строка содержит целое число k ($1 \leq k \leq n$).

Формат выходных данных

Если сортировку выполнить невозможно, выведите слово NO.

Иначе выведите слово YES и $2n$ строк, задающих программу сортировки. Каждая из этих строк должна описывать одну операцию и иметь следующий формат:

- $I(j)$ — считать элемент из входного потока и добавить его в очередь номер j ($1 \leq j \leq k$);
- $R(j)$ — извлечь элемент из очереди номер j ($1 \leq j \leq k$) и вывести его в выходной поток.

Входные данные

```
4
5 1 4 10
2
```

Выходные данные

YES

```
I(1)
I(2)
R(2)
I(2)
I(2)
R(2)
R(1)
R(2)
```

Входные данные

```
4
5 1 4 10
1
```

Выходные данные

NO