

## Задача 1. Двумерное домино

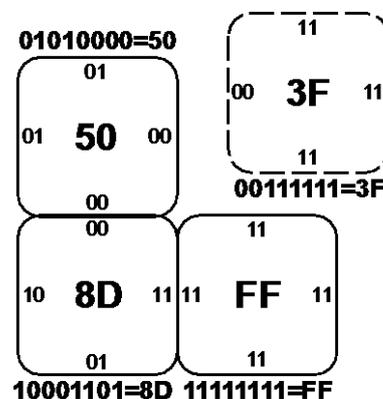
Входной файл: *input.txt*

Выходной файл: *output.txt*

Ограничение по времени: 1 секунда

Каждый элемент двумерного домино представляет собой квадрат, стороны которого несут на себе числа от 0 до 3, записанные в двоичной системе счисления. Домино считается составленным правильно, если каждая пара квадратов домино, имеющих общую сторону, несёт на этой стороне одно и то же число.

Необходимо определить, правильно ли составлено домино и можно ли добавить к нему ещё один заданный элемент так, чтобы он имел общую сторону хотя бы с одним из исходных квадратов. При добавлении элемент можно произвольно вращать, но не переворачивать.



### Входные данные

В первой строке входного файла через пробел размеры прямоугольника, заключающего в себе уже составленное домино:  $2 \leq N, M \leq 100$ .

Каждая из следующих  $N$  строк содержит через пробел описания  $M$  позиций в прямоугольнике в следующем виде:

- пустая позиция обозначается ZZ;
- элемент, занимающий непустую позицию, задаётся шестнадцатеричным текстовым представлением одного байта, в котором каждые два бита указывают число на одной из сторон. Стороны перечисляются по часовой стрелке.

Последняя строка содержит описание элемента, который предлагается добавить к уже составленному набору. Предполагается, что добавление происходит в пределах того же прямоугольника  $N \times M$ .

### Выходные данные

В выходном файле должно содержаться одно из трёх слов:

- ERROR (домино составлено неправильно);
- YES (домино составлено правильно, элемент добавить можно);
- NO (домино составлено правильно, но элемент добавить нельзя).

### Пример

**input.txt**

2 2

50 ZZ

8D FF

3F

**output.txt**

YES

## Задача 2. Задачка для папы

**Входной файл:** *input.txt*

**Выходной файл:** *output.txt*

**Ограничение по времени:** 1 секунда

Чтобы любознательный сын, школьник младших классов, не мешал смотреть после работы футбол, отец подсовывал ему разные задачки. Когда сын несколько подрос, он разгадал папину хитрость и решил отплатить ему тем же. Когда в очередной раз папа уселся перед телевизором, сын попросил помочь ему разгадать головоломку. Эту головоломку он сам и составил. Мальчик разложил несколько костяшек из комплекта домино в виде прямоугольника, а затем переписал на клеточный листок только количество очков с этих костяшек также в виде прямоугольника. Папе предстояло восстановить расположение костяшек домино на этом прямоугольнике.

### Входные данные

В первой строке входного файла через пробел размеры  $N$ ,  $M$  прямоугольника, составленного из пластинок домино,  $N$ ,  $M$  – натуральные числа, такие что  $N \times M$  – четное число от 2 до 56.

Каждая из следующих  $N$  строк содержит через пробел описания  $M$  значений, каждое из которых представляет собой целое число в диапазоне от 0 до 6 – количество очков на той квадратной части костяшки домино, которая оказалась в соответствующей позиции.

### Выходные данные

Если решение существует, то выходной файл содержит  $N$  строк, в каждой из которых через пробел  $M$  целых чисел – очки с квадратных частей костяшек домино. При этом если костяшка расположена вертикально, то перед количеством очков на ее частях необходимо ставить знак ‘-’.

Если решения не существует (сын ошибся при переписывании, и потому при любом разбиении прямоугольника на костяшки домино получается, что хотя бы одна костяшка встречается несколько раз), вывести ‘No solution’.

Если решение не единственное, то вывести любое из решений.

### Пример

**input.txt**

```
2 10
0 0 2 0 3 3 1 1 2 1
0 1 3 2 3 3 0 1 2 2
```

**output.txt**

```
-0 0 2 0 3 -3 1 1 -2 -1
-0 1 3 2 3 -3 0 1 -2 -2
```

### **Задача 3. Праздник**

**Входной файл:** *input.txt*

**Выходной файл:** *output.txt*

**Ограничение по времени:** 1 секунда

Одна фирма прославилась своим умением качественно организовывать праздники. Когда заказов стало очень много, работники фирмы ощутили явную нехватку времени и решили увеличить стоимость своих услуг: чем более срочным являлся заказ, тем выше становилась его цена. Для этого им понадобилось очень быстро определять количество рабочих дней между датой поступления заказа по интернету и датой празднования.

Составьте программу, которую фирма могла бы использовать для такого расчета, если считать, что в неделю сотрудники фирмы имеют только один выходной день – воскресенье, а в праздничные дни, не попавшие на воскресенье, работают. Сам день праздника в число рабочих дней до него, разумеется, не включается, в отличие от дня поступления заказа (если это не было воскресенье).

#### **Входные данные**

В первой строке через пробел три целых числа, определяющих дату праздника:  $D$ ,  $M$ ,  $Y$ , где  $Y$  – год ( $1980 \leq Y \leq 2020$ ),  $M$  – месяц ( $1 \leq M \leq 12$ ),  $D$  – число месяца  $M$ .

В первой строке через пробел три целых числа, определяющих дату поступления заказа:  $d$ ,  $m$ ,  $y$ , где  $y$  – год ( $1980 \leq y \leq 2020$ ),  $m$  – месяц ( $1 \leq m \leq 12$ ),  $d$  – число месяца  $m$ .

Гарантируется, что дата праздника всегда позже даты поступления заказа.

#### **Выходные данные**

В первой строке целое число – искомое число рабочих дней.

#### **Пример**

**input.txt**

18 5 2009

1 5 2009

**output.txt**

14

## Задача 4. Роботы

**Входной файл:** *input.txt*

**Выходной файл:** *output.txt*

**Ограничение по времени:** 2 секунды

На некотором заводе решили модернизировать производство и закупили для этого роботов. Так как для обработки детали требовалось выполнение двух операций, роботы также были двух типов: первую операцию выполняли роботы типа А, а вторую – роботы типа Б. Чтобы сэкономить на покупке роботов, было решено купить не новых роботов последней модели, а уже бывших в употреблении. В результате, время, которое разные роботы тратили на выполнение одной и той же операции, существенно различалось, что привело к трудностям в планировании работ.

Составьте программу, которая по заданному набору роботов обоих типов определяет, за какое минимальное время они смогут обработать определенное количество деталей.

### Входные данные

В первой строке натуральное число  $N$ ,  $1 \leq N \leq 100000$  – количество деталей, которое необходимо обработать.

Во второй строке натуральное число  $Na$ ,  $1 \leq Na \leq 1000$  – количество роботов, выполняющих первую операцию.

В третьей строке через пробел  $Na$  натуральных чисел  $A_i$ ,  $1 \leq A_i \leq 100$  – время, которое тратит  $i$ -ый робот типа А на выполнение операции.

В четвертой строке натуральное число  $Nb$ ,  $1 \leq Nb \leq 1000$  – количество роботов, выполняющих вторую операцию.

В пятой строке через пробел  $Nb$  натуральных чисел  $B_i$ ,  $1 \leq B_i \leq 100$  – время, которое тратит  $i$ -ый робот типа Б на выполнение операции.

### Выходные данные

В первой строке одно целое число – минимальное время, за которое все  $N$  деталей будут обработаны сначала роботом типа А, а потом роботом типа Б. Временем передачи детали от робота типа А роботу типа Б пренебречь.

### Пример

**input.txt**

6

3

1 3 2

2

2 3

**output.txt**

9

## Задача 5. Открытки

**Входной файл:** *input.txt*

**Выходной файл:** *output.txt*

**Ограничение по времени:** 1 секунда

Приближался Валентинов день. Маша с подружками решила этот праздник отметить и подготовить открытки. Для украшения открыток Маша использовала кружки разного размера и ажурную ленту. Технология изготовления открытки очень проста: сначала на картонную основу наклеиваются кружки, а затем по периметру полученной фигуры (или фигур, их может быть несколько) проклеивается лента так, чтобы закрыть все стыки кружков с картоном. Для этого Маше нужно вычислить длину ленты, нужную для украшения одной открытки. Помогите ей.

### Входные данные

Входной файл состоит из нескольких строк, каждая из которых задаёт кружок тремя вещественными числами:  $x$ ,  $y$ ,  $r$  через пробел ( $-10 \leq x, y \leq 10$ ,  $0.001 \leq r \leq 10$ ) с точностью до 3-х знаков после запятой. Количество кругов от 1 до 50.

### Выходные данные

Выходной файл содержит одно число – общую длину ленты для украшения открытки. Число выдается в формате с фиксированной точкой с тремя знаками после запятой.

#### Пример

**input.txt**

1 2 1

**output.txt**

6.283

**input.txt**

0 0 1.5

0 1 0.5

**output.txt**

9.425

## Задача 6. Симметрия

**Входной файл:** *input.txt*

**Выходной файл:** *output.txt*

**Ограничение по времени:** 1 секунда

Предприимчивая и умелая рукодельница решила подзаработать изготовлением «фенечек» из бисера. Любительница симметрии в искусстве, она использовала в своих орнаментах бусинки разных цветов (будем обозначать цвет целым положительным числом) по следующим правилам:

- 1) при длине ряда рисунка равной 1 использовала бусинку первого цвета;
- 2) при длине ряда рисунка равной 3 использовала бусинки двух цветов:  
1 2 1;
- 3) при необходимости добавления в рисунок еще одного цвета строился ряд:  
1 2 1 3 1 2 1 и так всякий раз в зависимости от числа используемых цветов, например, при использовании четырех цветов:  
1 2 1 3 1 2 1 4 1 2 1 3 1 2 1.

Напишите программу, которая помогла бы автоматизировать подбор цвета бусинки в ряду по её порядковому номеру.

### Входные данные

В первой строке целое число  $k$  ( $1 \leq k \leq 10^9$ ) – номер бусинки, цвет которой нужно определить, в ряду рисунка. Нумерация бусинок в ряду начинается с единицы.

### Выходные данные

В первой строке одно целое число – номер цвета заданной бусинки.

### Пример

**input.txt**

10

**output.txt**

2

## Задача 7. Ломоносов наших дней

**Входной файл:** *input.txt*

**Выходной файл:** *output.txt*

**Ограничение по времени:** 1 секунда

Вася Ломоносов проснулся и понял, что опаздывает на вступительный экзамен, до которого осталось ровно  $t$  часов. Он хотел добираться в университет из своей деревни пешком, как его великий предок, но теперь боялся опоздать и решил попросить помощи у колхозного шофера. Денег у Васи было немного, да и традицию хотелось соблюсти, поэтому он решил ехать как можно меньше. Он договорился с шофером, что тот довезёт его от деревни часть пути, а затем Вася пойдёт пешком. При поездке на автомобиле скорость Васи  $v_1$  км/час и оплата  $p_1$  рублей/км (округляется до ближайшего целого числа по стандартным правилам). Также Вася может идти пешком со скоростью  $v_2$  км/час. Денег он при этом, естественно, не тратит.

Может ли Вася успеть во время? Если да, то, какое минимальное количество денег ему при этом надо отдать шоферу? Придётся брать за калькулятор...

### Входные данные

В первой строке вводятся два числа:  $s$  и  $t$ ,  $1 \leq s \leq 1000$ ,  $1 \leq t \leq 50$ .  $s$  измеряется в км, а  $t$  – в часах.

Во второй строке вводятся три числа:  $v_1$ ,  $v_2$ ,  $p_1$ ,  $1 \leq v_1 \leq 100$ ,  $1 \leq v_2 \leq 15$ ,  $1 \leq p_1 \leq 100$ .

Все параметры целые числа.

### Выходные данные

Если возможно успеть вовремя, то в выходной строке вывести одно целое число – минимальное количество денег, которое нужно потратить (в рублях). В противном случае вывести одно слово “NO”.

### Примеры

<b>input.txt</b>	<b>output.txt</b>
14 2 10 4 10	100
154 2 75 4 10	NO

## Задача 8. Небоскрёбы

**Входной файл:** *input.txt*

**Выходной файл:** *output.txt*

**Ограничение по времени:** 2 секунды

Под конец семестра студент архитектурной академии Петя узнал, что ему срочно нужно сдать курсовой проект, задание к которому звучало следующим образом: на квадратной площадке размером  $N \times N$ , поделенной на единичные квадратики, разместить дома высотой от 1 до  $N$  этажей, по одному в каждом квадратике так, чтобы в каждой строке и в каждом столбце этажность всех домов была различна.

Поняв, что сам он справиться с заданием в срок не успеет, Петя поспрашивал у старших товарищей, не завалилось ли у них черновиков с подобным заданием. К счастью, один из старшекурсников поделился с Петей своими результатами, но выглядели они весьма странно. Для строк и столбцов площадки было подписано, сколько зданий будет видно, если смотреть на них слева или справа (для строк) и сверху или снизу (для столбцов). Здание считается видимым, если перед ним нет зданий большей этажности. Более того, так как черновики были довольно старыми, некоторые подписи стерлись.

Помогите Пете по заданному черновику восстановить хотя бы один из вариантов расположения домов, подходящий под это описание.

### Входные данные

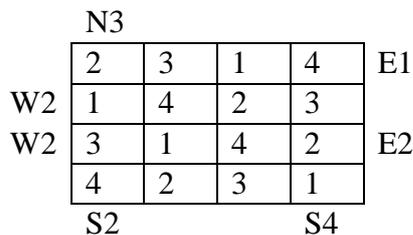
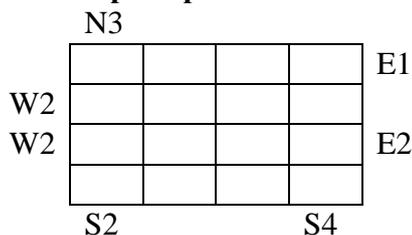
В первой строке два натуральных числа через пробел:  $N$ ,  $2 \leq N \leq 4$  – размер квадратной площадки, а также максимальная этажность зданий, и  $K$  – количество подписей, сохранившихся в черновике,  $0 \leq K \leq N \cdot 4$ .

В последующих  $K$  строках записаны подписи из черновика в следующем формате: номер строки или столбца, пробел, направление ( $W$  – строка слева направо,  $E$  – строка справа налево,  $N$  – столбец сверху вниз,  $S$  – столбец снизу вверх), пробел, количество видимых домов в этом направлении. Столбцы нумеруются слева направо, а строки – сверху вниз. Нумерация начинается с единицы.

### Выходные данные

В выходной файл выводится  $N$  строк. В каждой строке  $N$  чисел через пробел – этажности зданий, расположенных в соответствующей строке площадки в порядке увеличения номера столбца. Результирующее расположение домов должно удовлетворять подписям с черновика. Если вариантов расположения домов несколько, то вывести любой из них. В случае отсутствия решения вывести 'No solution'.

### Пример



### input.txt

```
4 7
1 N 3
2 S 2
4 S 4
1 E 1
2 W 2
3 E 2
3 W 2
```

### output.txt

```
2 3 1 4
1 4 2 3
3 1 4 2
4 2 3 1
```

## Задача 9. Пробки на дорогах

**Входной файл:** *input.txt*

**Выходной файл:** *output.txt*

**Ограничение по времени:** 2 секунды

В городе N все дороги двусторонние. Система дорог города N позволяет проехать из каждой точки города в любую другую, но из-за стремительного увеличения числа автомобилей постоянно возникают пробки, проехать через которые за разумное время практически невозможно. По настойчивой просьбе автомобилистов мэрия создала информационную службу, которая сообщает автомобилистам о том, как добраться из одной точки города в другую кратчайшим путем, то есть, проехав наименьшее возможное число перекрестков, минуя возникшие пробки. Напишите программу, которая позволит автоматически прокладывать этот кратчайший путь, облегчая жизнь как автомобилистам, так и сотрудникам службы.

### Входные данные

В первой строке через пробел три целых числа:  $n$ ,  $m$ ,  $k$  ( $3 \leq n$ ,  $m \leq 1000$ ,  $1 \leq k \leq 50$ );  $n$  – количество перекрестков,  $m$  – количество дорог,  $k$  – количество запросов по расчету оптимального маршрута. Нумерация перекрестков, дорог и автомобилей начинается с единицы. Следующие  $m$  строк задают список дорог в городе, в каждой строке пара чисел от 1 до  $n$  через пробел – номера перекрестков, соединенных дорогой. Далее  $k$  блоков, каждый из которых описывает один запрос. Блок начинается строкой, содержащей три целых числа, записанных через пробел:  $s$ ,  $f$ ,  $p$  ( $1 \leq s, f, p \leq m$ ),  $s$ ,  $f$  – номера дорог, являющихся начальным и целевым пунктом движения автомобиля соответственно,  $p$  – количество пробок. В следующих  $p$  строках по одному числу от 1 до  $m$  – номера дорог, на которых возникли пробки.

### Выходные данные

Выходной файл содержит  $k$  блоков, описывающих маршрут движения автомобиля с минимальным количеством пройденных перекрестков. В первой строке блока выводится целое число – количество перекрестков, через которые проходит маршрут. Во второй строке маршрут – последовательность номеров перекрестков (целых чисел от 1 до  $n$ ) через пробел. Гарантируется, что маршрут из начального пункта в конечный всегда существует.

### Пример

**input.txt**

7 8 2

1 2

2 3

3 4

4 5

5 6

6 7

1 7

1 5

7 4 1

8

1 5 1

2

**output.txt**

3

7 6 5

2

1 5