

Задача 1. Драгоценные камни (100 баллов)

Имя входного файла: gems.in
Имя выходного файла: gems.out
Максимальное время работы на одном тесте: 1 секунда
Максимальный объем используемой памяти: 64 Кб

Гномы, жители страны Мории, добывают драгоценные камни трех видов — поделочные (вид А), полудрагоценные (вид В) и драгоценные (вид С). Хранят их гномы в одинаковых коробках, каждую из которых наполняют только одним видом камней. Заполненные коробки гномы складывают в огромных залах Мории одну на другую так, что получаются вертикальные стопки.

Самые «плохие» стопки гномы отдают оркам. Стопка считается плохой, если в ней подряд лежат более одной коробки, заполненной камнями вида А. Стопка считается «хорошей», если она не является «плохой».

От Вас **требуется** написать программу, которая для заданного количества коробок N будет определять число возможных «хороших» стопок.

Формат входного файла

В единственной строке входного файла записано одно натуральное число $1 \leq N \leq 20$.

Формат выходного файла

В единственной строке входного файла записано одно натуральное число — количество «хороших» вариантов формирования стопки.

Пример входного и выходного файлов

gems.in	gems.out
2	8

Примечание к примеру:

В примере из условия среди стопок длины 2 бывают «хорошие» стопки видов АВ, АС, ВА, ВВ, ВС, СА, СВ и СС. Стопки типа АА являются «плохими».

Задача 2. Город будущего (100 баллов)

Имя входного файла: city.in
Имя выходного файла: city.out
Максимальное время работы на одном тесте: 1 секунда
Максимальный объем используемой памяти: 64 Кб

В городе будущего все дома – это небоскребы. Всего в городе M улиц, которые пересекаются и дают всего N перекрестков. Для организации эффективного движения необходимо, чтобы перед каждым перекрестком

вдоль улицы было размещено лазерное устройство, отслеживающее поток транспортных средств, движущихся к перекрестку. Каждое такое устройство (лазер) работает только в одну сторону – от перекрестка вдоль улицы. Движение по всем улицам двустороннее. Любые два перекрестка соединены только одной улицей. Нет улиц от k -го перекрестка до него самого.

Итак, от Вас **требуется** написать программу, вычисляющую для каждого перекрестка количество лазеров, которые необходимо на нем поставить властям города.

Формат входных данных

Первая строка входного файла содержит два натуральных числа N и M ($0 < N \leq 100$, $0 \leq M \leq N*(N-1)/2$).

Каждая из следующих M строк содержит разделенные пробелом два натуральных числа i и j ($1 \leq i, j \leq N$), которые означают, что перекрестки i и j соединены улицей.

Формат выходных данных

В первой строке выходного файла нужно вывести N чисел: i -ое число означает количество лазеров на i -ом перекрестке.

Пример входного и выходного файлов

city.in	city.out
7 10	3 3 2 2 5 2 3
5 1	
3 2	
7 1	
5 2	
7 4	
6 5	
6 4	
7 5	
2 1	
5 3	

Задача 3. Площадка перед домом (100 баллов)

Имя входного файла:	tile.in
Имя выходного файла:	tile.out
Максимальное время работы на одном тесте:	2 секунды
Максимальный объем используемой памяти:	64 КБ

Иван Петрович купил для укладки дорожек на своем участке некоторое количество тротуарной плитки. После того, как все дорожки были готовы, осталось еще n плиток одинакового размера: w — ширина и h — высота. Иван Петрович решил уложить на своем участке перед домом еще и квадратную площадку так, чтобы она занимала как можно меньше

места на участке. Для того, чтобы площадка была красивая, каждая плитка должна быть размещена строго в прямоугольнике размером w на h . Плитка нельзя поворачивать на 90 градусов и размещать так, чтобы они накладывались одна на другую, а то площадка будет с ухабами.

От Вас **требуется** написать программу, которая вычислит минимальный размер стороны площадки, которая получится перед домом Ивана Петровича.

Формат входных данных

Входной файл содержит одну строку, в которой находятся разделенный пробелом три натуральных числа: w, h, n ($1 \leq w \cdot h \cdot n \leq 10^9$).

Формат выходных данных

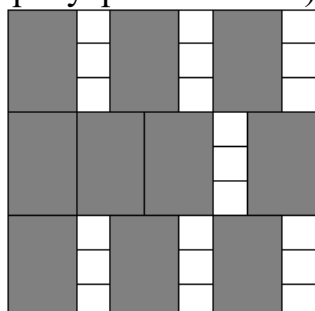
Одно натуральное число – минимальный размер стороны площадки.

Примеры входных и выходных файлов

tile.in	tile.out
2 3 10	9

tile.in	tile.out
1 1 1	1

Рисунок, поясняющий первый пример (закрашенные прямоугольники – это тротуарные плитки):



Задача 4. Сложный шифр (100 баллов)

Имя входного файла:	cipher.in
Имя выходного файла:	cipher.out
Максимальное время работы на одном тесте:	1 секунда
Максимальный объем используемой памяти:	64 Кб

Для передачи сообщения по открытому каналу связи шифровальщики придумали сложный шифр. Шифр состоит из N различных чисел, записанных в одну строку через пробелы. Дешифрованное сообщение состоит из K чисел, которые являются номерами исходных чисел, разбивающих исходное сообщение на K частей (подпоследовательностей), так что сумма минимумов каждой подпоследовательности будет максимальной.

Итак, от Вас **требуется** написать программу, которая выполняет дешифровку заданного шифра.

Формат входных данных

Первая строка входного файла содержит два разделенных пробелом натуральных числа N и K ($1 \leq K < N \leq 300$).

Вторая строка входного файла содержит зашифрованное сообщение – N разделенных пробелом целых различных натуральных чисел $a_1, a_2, a_3 \dots a_n$ ($1 \leq a_i \leq 10^9$).

Формат выходных данных

Выходной файл в первой строке содержит максимальную сумму K слагаемых, которые являются минимумами каждой части разбиения последовательности на K частей.

Во второй строке выходного файла содержится дешифрованное сообщение из разделенных пробелом K натуральных чисел – номеров чисел в последовательности, которые завершают каждую из подпоследовательностей зашифрованного сообщения.

Пример входного и выходного файлов

cipher.in	cipher.out
10 5	27
1 10 2 8 9 3 5 4 7 6	3 4 5 8 10

Пояснения к примеру::

В примере зашифрованное сообщение из 10 чисел разбивается на 5 частей:

{1, 10, 2}, {8}, {9}, {3 5 4}, {7 6}

Искомая сумма равна $1+8+9+3+6=27$