

## Задача А. Скобочные последовательности

Имя входного файла: `brackets.in`  
Имя выходного файла: `brackets.out`  
Ограничение по времени: 1 секунда  
Ограничение по памяти: 64 мегабайта

Правильная скобочная последовательность (ПСП) — строка, заданная следующим определением:

- Пустая строка является ПСП;
- Если строки  $S$  и  $T$  являются ПСП, то строки  $(S)$  и  $ST$  также являются ПСП;
- Никакая другая строка не является ПСП.

Дано целое положительное число  $n$ . Определите количество правильных скобочных последовательностей, состоящих из  $2n$  скобок.

Например, существует 5 правильных скобочных последовательностей длины 6:  
 $((())), ((())), ((())(), ()()), ()()$ .

### Формат входного файла

Во входном файле содержится целое положительное число  $n$  ( $1 \leq n \leq 18$ ).

### Формат выходного файла

Выведите количество правильных скобочных последовательностей длины  $2n$ .

### Пример

<code>brackets.in</code>	<code>brackets.out</code>
3	5

## Задача В. Калькулятор

Имя входного файла: `calcul.in`  
Имя выходного файла: `calcul.out`  
Ограничение по времени: 1 секунда  
Ограничение по памяти: 64 мегабайта

Имеется калькулятор, который выполняет следующие операции:

- Умножить число  $X$  на 2.
- Умножить число  $X$  на 3.
- Прибавить к числу  $X$  единицу.

Определите, какое наименьшее количество операций требуется, чтобы получить из числа 1 число  $N$ .

### Формат входного файла

Во входном файле написано натуральное число  $N$ , не превосходящее  $10^6$ .

### Формат выходного файла

В первой строке выходного файла выведите минимальное количество операций. Во второй строке выведите числа, последовательно получающиеся при выполнении операций. Первое из них должно быть равно 1, а последнее  $N$ .

### Пример

<code>calcul.in</code>	<code>calcul.out</code>
1	0 1
5	3 1 2 4 5
962340	17 1 3 9 27 54 55 165 495 1485 4455 8910 17820 17821 53463 160389 160390 481170 962340

## Задача С. Диаметр графа

Имя входного файла: `diameter.in`  
Имя выходного файла: `diameter.out`  
Ограничение по времени: 1 секунда  
Ограничение по памяти: 64 мегабайта

Дан связный взвешенный неориентированный граф.

Рассмотрим пару вершин, расстояние между которыми максимально среди всех пар вершин. Расстояние между ними называется *диаметром графа*. *Эксцентриситетом вершины* называется максимальное расстояние от вершины  $v$  до других вершин графа. *Радиусом графа* называется наименьший из эксцентриситетов вершин. Найдите диаметр и радиус графа.

### Формат входного файла

В первой строке входного файла единственное число:  $N$  ( $1 \leq N \leq 100$ ) — количество вершин графа. В следующих  $N$  строках по  $N$  чисел — матрица смежности графа, где  $-1$  означает отсутствие ребра между вершинами, а любое

неотрицательное число — присутствие ребра данного веса. На главной диагонали матрицы всегда нули; веса рёбер не превышают 1000.

### Формат выходного файла

В выходной файл выведите два числа - диаметр и радиус графа.

### Пример

diameter.in	diameter.out
4	8
0 -1 1 2	5
-1 0 -1 5	
1 -1 0 4	
2 5 4 0	

## Задача D. Расстояние между вершинами

Имя входного файла: `distance.in`

Имя выходного файла: `distance.out`

Ограничение по времени: 1 секунда

Ограничение по памяти: 64 мегабайта

Дан неориентированный взвешенный граф. Найти вес минимального пути между двумя вершинами.

### Формат входного файла

Первая строка входного файла содержит натуральные числа  $N, M, S$  и  $F$  ( $N \leq 5000, M \leq 100000, 1 \leq S, F \leq N, S \neq F$ ) — количество вершин и ребер графа а также номера вершин, длину пути между которыми требуется найти. Следующие  $M$  строк по три натуральных числа  $b_i, e_i$  и  $w_i$  — номера концов  $i$ -ого ребра и его вес соответственно ( $1 \leq b_i, e_i \leq n, 0 \leq w_i \leq 100000$ ).

### Формат выходного файла

Первая строка должна содержать одно натуральное число — вес минимального пути между вершинами  $S$  и  $F$ . Во второй строке через пробел выведите вершины на кратчайшем пути из  $S$  в  $F$  в порядке обхода.

Если путь из  $S$  в  $F$  не существует, выведите  $-1$ .

### Пример

distance.in	distance.out
4 4 1 3	3
1 2 1	1 2 3
3 4 5	
3 2 2	
4 1 4	

## Задача E. Факториал

Имя входного файла: `fact.in`

Имя выходного файла: `fact.out`

Ограничение по времени: 1 секунда

Ограничение по памяти: 64 мегабайта

### Формат входного файла

Во входном файле одно целое число  $n$  ( $0 \leq n \leq 3000$ ).

### Формат выходного файла

В выходной файл выведите одно число, равное факториалу числа  $n$ .

### Пример

fact.in	fact.out
3	6

## Задача F. Ход конём - 2

Имя входного файла: `knight2.in`

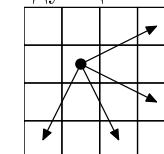
Имя выходного файла: `knight2.out`

Ограничение по времени: 1 секунда

Ограничение по памяти: 64 мегабайта

Дана прямоугольная доска  $N \times M$  ( $N$  строк и  $M$  столбцов). В левом верхнем углу находится шахматный конь, которого необходимо переместить в правый нижний угол доски.

При этом конь может ходить следующим образом:



Необходимо определить, сколько существует различных маршрутов, ведущих из левого верхнего в правый нижний угол.

#### Формат входного файла

В первой строке входного файла находятся два натуральных числа  $N$  и  $M$  ( $1 \leq N, M \leq 50$ ).

#### Формат выходного файла

В выходной файл выведите единственное число — количество способов добраться конём до правого нижнего угла доски.

#### Пример

knights.in	knights.out
4 4	2
15 14	7884330

### Задача G. Разбиения на слагаемые

Имя входного файла: partition.in  
Имя выходного файла: partition.out  
Ограничение по времени: 1 секунда  
Ограничение по памяти: 64 мегабайта

Перечислите все разбиения целого положительного числа  $N$  на целые положительные слагаемые. Разбиения должны обладать следующими свойствами:

1. Слагаемые в разбиениях идут в невозрастающем порядке.
2. Разбиения перечисляются в лексикографическом порядке.

#### Формат входного файла

Во входном файле находится единственное число  $N$  ( $1 \leq N \leq 40$ ).

#### Формат выходного файла

Перечислить искомые разбиения.

#### Пример

partition.in	partition.out
4	1 1 1 1 2 1 1 2 2 3 1 4

### Задача H. Перестановки

Имя входного файла: perm.in  
Имя выходного файла: perm.out  
Ограничение по времени: 1 секунда  
Ограничение по памяти: 64 мегабайта

Дана перестановка из  $N$  чисел от 1 до  $N$ . Требуется найти её номер в лексикографическом порядке.

#### Формат входного файла

Во входном файле сначала записано число  $N$  ( $1 \leq N \leq 12$ ). В следующей строке записана сама перестановка —  $N$  чисел, разделённых пробелами.

#### Формат выходного файла

В выходной файл нужно вывести единственное число — номер перестановки в лексикографическом порядке.

#### Пример

perm.in	perm.out
3 2 1 3	3

### Задача I. Принадлежность точки отрезку

Имя входного файла: point3.in  
Имя выходного файла: point3.out  
Ограничение по времени: 1 секунда  
Ограничение по памяти: 64 мегабайта

#### Формат входного файла

Шесть чисел — координаты точки и координаты концов отрезка.

#### Формат выходного файла

Одна строка «YES», если точка принадлежит отрезку, и «NO» в противном случае.

#### Пример

point3.in	point3.out
3 3 1 2 5 4	YES

## Задача J. Выпуклый многоугольник

Имя входного файла: `polygon.in`  
Имя выходного файла: `polygon.out`  
Ограничение по времени: 1 секунда  
Ограничение по памяти: 64 мегабайта

### Формат входного файла

В первой строке одно число  $N$  ( $3 \leq N \leq 100000$ ). Далее в  $N$  строках по паре целых чисел — координаты очередной вершины простого многоугольника в порядке обхода по или против часовой стрелки.

### Формат выходного файла

Одна строка «YES», если приведённый многоугольник является выпуклым, и «NO» в противном случае.

### Пример

<code>polygon.in</code>	<code>polygon.out</code>
3 1 0 0 1 1 1	YES