

## Задача А. Пути на доске

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

Рассмотрим бесконечную клетчатую доску.

Назовём *путём* из одной клетки в другую последовательность клеток, в которой каждые две идущие подряд клетки являются соседними по стороне. Длина пути — это количество клеток в нём, не считая начальную.

Назовём путь *простым*, если в нём не встречается двух одинаковых клеток.

Зафиксируем какую-то клетку на доске. Сколько существует простых путей заданной длины, начинающихся в этой клетке?

### Формат входного файла

В первой строке входного файла задано целое число  $n$  ( $0 \leq n \leq 22$ ).

### Формат выходного файла

В первой строке выходного файла выведите одно число — количество путей длины  $n$  из этой клетки.

### Примеры

| <code>paths.in</code> | <code>paths.out</code> |
|-----------------------|------------------------|
| 0                     | 1                      |
| 1                     | 4                      |
| 2                     | 12                     |

## Задача В. Самый длинный путь

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

В данном ориентированном графе найдите самый длинный путь такой, что каждая вершина графа встречается в нём не более одного раза.

### Формат входного файла

В первой строке входного файла заданы через пробел два целых числа  $n$  и  $m$  ( $1 \leq n \leq 22$ ,  $0 \leq m \leq 1000$ ). В следующих  $m$  строках заданы рёбра графа в формате  $u_i v_i$  — номера начальной и конечной вершин ребра  $i$ , соответственно. Граф может содержать петли и кратные рёбра.

### Формат выходного файла

В первой строке выходного файла выведите длину искомого пути  $l$ . Во второй строке выведите  $l + 1$  число через пробел — вершины пути в порядке обхода. Если оптимальных ответов несколько, можно вывести любой из них.

### Примеры

| <code>path.in</code>                          | <code>path.out</code> |
|---|-----------------------|
| 3 3<br>1 2<br>2 3<br>3 1                      | 2<br>1 2 3            |
| 4 6<br>1 2<br>2 1<br>2 3<br>2 4<br>3 2<br>4 2 | 2<br>1 2 4            |
| 5 3<br>3 2<br>2 2<br>1 5                      | 1<br>3 2              |

## Задача С. Сортировка вручную

Имя входного файла: `sort.in`  
Имя выходного файла: `sort.out`  
Ограничение по времени: 2.5 секунды  
Ограничение по памяти: 64 мегабайта

Пришло время расставить книжки на полке. На ней стоят  $n$  книг, однозначно пронумерованных целыми числами от 1 до  $n$ . Требуется расставить их в порядке возрастания номеров.

Вы можете сортировать книги, вставляя  $i$ -ю книгу на  $i$ -ю позицию. Сколько операций вам придётся выполнить для успешной сортировки?

### Формат входного файла

В первой строке входного файла содержится целое число  $n$  ( $1 \leq n \leq 20$ ) — количество книг на полке.

Во второй строке записано  $n$  различных целых чисел от 1 до  $n$  — номера книг в том порядке, в котором они расставлены вначале.

### Формат выходного файла

Выведите единственное целое число — минимальное количество операций указанного вида, которое придётся выполнить для сортировки.

### Примеры

| <code>sort.in</code>   | <code>sort.out</code> |
|------------------------|-----------------------|
| 3<br>1 2 3             | 0                     |
| 3<br>2 1 3             | 1                     |
| 3<br>3 2 1             | 2                     |
| 9<br>7 2 9 4 6 8 5 3 1 | 5                     |

## Задача D. Сокровища

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

Дочь короля Флатландии собирается выйти за прекрасного принца. Принц хочет подарить принцессе сокровища, но он не уверен какие именно бриллианты из своей коллекции выбрать.

В коллекции принца  $n$  бриллиантов, каждый характеризуется весом  $w_i$  и стоимостью  $v_i$ . Принц хочет подарить наиболее дорогие бриллианты, однако король умен и не примет бриллиантов суммарного веса больше  $R$ . С другой стороны, принц будет считать себя жадным всю оставшуюся жизнь, если подарит бриллиантов суммарным весом меньше  $L$ .

Помогите принцу выбрать набор бриллиантов наибольшей суммарной стоимости, чтобы суммарный вес был в отрезке  $[L, R]$ .

### Формат входного файла

Первая строка содержит число  $n$  ( $1 \leq n \leq 32$ ),  $L$  и  $R$  ( $0 \leq L \leq R \leq 10^{18}$ ). Следующие  $n$  строк описывают бриллианты и содержат по два числа — вес и стоимость соответствующего бриллианта ( $1 \leq w_i, v_i \leq 10^{15}$ ).

### Формат выходного файла

Первая строка вывода должна содержать  $k$  — количество бриллиантов, которые нужно подарить королю. Вторая строка должна содержать номера даримых бриллиантов.

Бриллианты нумеруются от 1 до  $n$  в порядке появления во входных данных.

Если составить подарок королю невозможно, то выведите 0 в первой строке вывода.

### Пример

| <code>dowry.in</code> | <code>dowry.out</code> |
|-----------------------|------------------------|
| 3 6 8                 | 1                      |
| 3 10                  | 2                      |
| 7 3                   |                        |
| 8 2                   |                        |