

## Задача А. Перестановки

Имя входного файла: `permutations.in`  
Имя выходного файла: `permutations.out`  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 64 мегабайта

Выведите в выходной файл все перестановки из  $n$  чисел.

### Формат входного файла

Во входном файле задано число  $n$  ( $1 \leq n \leq 8$ ).

### Формат выходного файла

Выведите в выходной файл в лексикографическом порядке все перестановки чисел от 1 до  $n$ .

### Пример

<code>permutations.in</code>	<code>permutations.out</code>
3	1 2 3 1 3 2 2 1 3 2 3 1 3 1 2 3 2 1

## Задача В. Оптимальная перестановка

Имя входного файла: `optimal.in`  
Имя выходного файла: `optimal.out`  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 64 мегабайта

Будем называть ценностью перестановки  $\langle a_1, a_2, \dots, a_n \rangle$  величину  $(a_1 a_2 + a_2 a_3 + \dots + a_{n-1} a_n) \bmod r$ .

Вам даны  $n$  и  $r$ , найдите перестановку, имеющую максимальную ценность. Если таких перестановок несколько, выведите первую лексикографически.

### Формат входного файла

Во входном файле заданы числа  $n$  и  $r$  ( $2 \leq n \leq 10$ ,  $2 \leq r \leq 100$ ).

### Формат выходного файла

Выведите в выходной файл две строки. Первая строка должна содержать максимальное возможное значение ценности перестановки. Вторая должна содержать оптимальную перестановку. Если таких перестановок несколько, выведите первую в лексикографическом порядке.

### Пример

<code>optimal.in</code>	<code>optimal.out</code>
3 5	4 1 3 2

## Задача С. Красивые перестановки

Имя входного файла: `beautiful.in`  
Имя выходного файла: `beautiful.out`  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 64 мегабайта

Будем называть ценностью перестановки  $\langle a_1, a_2, \dots, a_n \rangle$  величину  $(a_1a_2 + a_2a_3 + \dots + a_{n-1}a_n) \bmod r$ .

Петя считает число красивым, если оно либо равно 0, либо число его делителей кратно 3. Например, 9 — красивое число (у него три делителя: 1, 3 и 9), а 6 — нет (у него 4 делителя: 1, 2, 3 и 6).

Вам даны  $n$  и  $r$ , найдите число перестановок, ценность которых является красивым числом.

### Формат входного файла

Во входном файле заданы числа  $n$  и  $r$  ( $2 \leq n \leq 10$ ,  $2 \leq r \leq 1000$ ).

### Формат выходного файла

Выведите в выходной файл число перестановок, ценность которых является красивым числом.

### Пример

<code>beautiful.in</code>	<code>beautiful.out</code>
3 10	2

### Комментарий

В примере искомые перестановки —  $\langle 1, 3, 2 \rangle$  и  $\langle 2, 3, 1 \rangle$

## Задача D. Двоичные вектора

Имя входного файла: `vectors.in`  
Имя выходного файла: `vectors.out`  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 64 мегабайта

Выведите в выходной файл все двоичные вектора.

### Формат входного файла

Во входном файле задано число  $n$  ( $1 \leq n \leq 10$ ).

### Формат выходного файла

В первой строке выходного файла выведите количество двоичных векторов длины  $n$ . В следующих строках выведите сами эти вектора в лексикографическом порядке по одному в строке.

### Пример

<code>vectors.in</code>	<code>vectors.out</code>
3	8 000 001 010 011 100 101 110 111

### Система оценивания

Только 100. Ну, или 0.

## Задача Е. Двоичные вектора 2

Имя входного файла: `vectors2.in`  
Имя выходного файла: `vectors2.out`  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 64 мегабайта

Выведите в выходной файл все двоичные вектора, в которых нет двух единиц подряд.

### Формат входного файла

Во входном файле задано число  $n$  ( $1 \leq n \leq 16$ ).

### Формат выходного файла

В первой строке выходного файла выведите количество двоичных векторов длины  $n$  в которых нет двух единиц подряд. В следующих строках выведите сами эти вектора в лексикографическом порядке по одному в строке.

### Пример

<code>vectors2.in</code>	<code>vectors2.out</code>
3	5 000 001 010 100 101

## Задача F. Сочетания

Имя входного файла: `choose.in`  
Имя выходного файла: `choose.out`  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 64 мегабайта

Выведите все сочетания из  $n$  по  $k$ .

### Формат входного файла

Во входном файле заданы числа  $n$  и  $k$ .

### Формат выходного файла

Выведите в выходной файл все сочетания по  $k$  из чисел от 1 до  $n$  в лексикографическом порядке.  $1 \leq k \leq n \leq 100$ , число сочетаний, которое необходимо вывести, не превосходит 100 000. Суммарное количество элементов в этих сочетаниях не превышает  $10^6$ .

### Пример

<code>choose.in</code>	<code>choose.out</code>
4 2	1 2 1 3 1 4 2 3 2 4 3 4

## Задача G. Останки Юрского периода

Имя входного файла: `jurassic.in`  
Имя выходного файла: `jurassic.out`  
Ограничение по времени: 2 seconds  
Ограничение по памяти: 256 мегабайта

Археологи недавно нашли фрагменты динозавра Юрского периода. Археологи решили, что они отправят кости динозавра в музей. Но, к сожалению, кости такие большие, что они не смогли положить их в один ящик. Поэтому они разделили скелет на отдельные кости и отправили каждый из них по-отдельности. Теперь сотрудникам музея предстоит собрать скелет. Для того, чтобы они могли это сделать, археологи отметили точки, в которых кости должны были быть соединены, специальными пометками. А именно — в каждой точке, где соединялись две кости, археологи написали на каждой из них одинаковые заглавные латинские буквы.

Однако пока археологи разбирали и упаковывали скелет, были обнаружены еще кости и они тоже были отправлены в музей. Причем они тоже могли быть соединены друг с другом, поэтому на них также могли быть пометки. Более того, археологи всегда использовали одинаковые пометки в точках на костях, которые должны были быть соединены, но они могли использовать одну и ту же пометку для различных соединений. Правда на каждой кости было не более одной пометки конкретной буквой.

Теперь работники музея пытаются разобраться с этой ситуацией и найти хотя бы какое-нибудь теоретически возможное множество костей исходного скелета динозавра. А именно, они хотят найти множество костей, которое удовлетворяет следующим условиям.

- Кости, помеченные некоторой буквой, можно попарно соединить друг с другом. Иначе говоря, каждая пометка встречается четное число раз.
- Число костей в наборе максимально.

### Формат входного файла

Первая строка входного файла содержит  $N$  — число костей ( $1 \leq N \leq 24$ ). Следующие  $N$  строк содержат описание костей: каждая строка содержит непустую последовательность различных заглавных букв латинского алфавита — метки на костях.

### Формат выходного файла

На первой строке выходного файла выведите  $L$  — максимальное возможное число костей, из которых можно собрать скелет. Вторая строка должна содержать  $L$  чисел — номера костей. Кости пронумерованы от 1 до  $N$  в порядке, в котором они заданы во входном файле.

### Пример

jurassic.in	jurassic.out
6 ABD EG GE ABE AC BCD	5 1 2 3 5 6
1 ABC	0