

## Задача G. Anti-qsort test

Имя входного файла: стандартный ввод  
Имя выходного файла: стандартный вывод  
Ограничение по времени: 1 секунда  
Ограничение по памяти: 256 мегабайт

Рассмотрим алгоритм быстрой сортировки, с выбором в качестве барьерного элемента среднего элемента на отрезке ( $q = a[(l + r) / 2]$ ):

```
void qsort(vector<int> & a, int left, int right)
// Сортировка a[left...right] включительно
{
    if (right <= left)
        return;
    int q = a[(l + r) / 2];
    int i = left;
    int j = right;
    while (i <= j) {
        while (a[i] < q)
            ++i;
        while (q < a[j])
            --j;
        if (i <= j) {
            swap(a[i], a[j]);
            ++i;
            --j;
        }
    }
    qsort(a, left, j);
    qsort(a, i, right);
}
```

По данному числу  $n$  составьте тест, являющийся перестановкой чисел от 1 до  $n$ , на котором этот алгоритм выполняет наибольшее число сравнений (подсчитываются сравнения  $a[i] < q$  и  $q < a[j]$ ).

### Формат входных данных

Программа получает на вход одно целое число  $n$ ,  $1 \leq n \leq 70\,000$ .

### Формат выходных данных

Программа должна вывести перестановку чисел от 1 до  $n$ , на которой данная реализация алгоритма быстрой сортировки Хоара будет выполнять наибольшее число сравнений.

Можно вывести любой из возможных ответов.

### Пример

стандартный ввод	стандартный вывод
3	1 3 2