

Задача D. Анти-QuickSort (2 балла)

Имя входного файла: **antiqs.in**
Имя выходного файла: **antiqs.out**
Ограничение по времени: **2 секунды**
Ограничение по памяти: **256 мегабайт**

Для сортировки последовательности чисел широко используется быстрая сортировка — QuickSort. Далее приведена программа, которая сортирует массив **a**, используя этот алгоритм.

```
var a : array [1..N] of integer;  
  
procedure QSort(left , right : integer);  
var i, j, key, buf : integer;  
begin  
  key := a[(left + right) div 2];  
  i := left;  
  j := right;  
  repeat  
    while a[i] < key do        {первый while}  
      inc(i);  
    while key < a[j] do        {второй while}  
      dec(j);  
    if i <= j then begin  
      buf := a[i];  
      a[i] := a[j];  
      a[j] := buf;  
      inc(i);  
      dec(j);  
    end;  
    until i > j;  
    if left < j then QSort(left , j);  
    if i < right then QSort(i , right);  
end;  
  
begin  
  ...  
  QSort(1 , N);  
end.
```

Хотя QuickSort является самой быстрой сортировкой в среднем, существуют тесты, на которых она работает очень долго. Оценивать время работы алгоритма будем количеством сравнений с элементами массива (то есть суммарным количеством сравнений в первом и втором **while**). Требуется написать программу, генерирующую тест, на котором быстрая сортировка сделает наибольшее число таких сравнений.

Формат входного файла

В первой строке находится единственное число n ($1 \leq n \leq 70000$).

Формат выходного файла

Вывести перестановку чисел от 1 до n , на которой быстрая сортировка выполнит максимальное число сравнений. Если таких перестановок несколько, вывести любую из них.

Пример

antiqs.in	antiqs.out
3	1 3 2