QSort. Интеллект против QSort

Имя входного файла: `qsort.in`
Имя выходного файла: `qsort.out`

Для сортировки последовательности чисел широко используется быстрая сортировка — QSort. Ниже приведена программа, которая сортирует массив a, используя данный алгоритм.

```
var a : array [1..N] of integer;

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

Хотя QSort является самой быстрой сортировкой в среднем, существуют тесты, на которых она работает очень долго. Будем оценивать время работы алгоритма количеством сравнений, сделанных с элементами массива (т.е. суммарным количеством сравнений в первой и втором while). Требуется написать программу, генерирующую тест, на котором быстрая сортировка сделает наибольшее число таких сравнений.

Формат входного файла

Во входном файле задано число N ($1 \le N \le 70\,000$).

Формат выходного файла

В выходной файл вывести перестановку чисел от 1 до N, на которой быстрая сортировка выполнит максимальное число сравнений. Если таких перестановок несколько, вывести любую из них.

Пример

qsort.in	qsort.out
3	1 3 2