

Задача А. Сортировка

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

Дан массив целых чисел. Ваша задача — отсортировать его в порядке неубывания. Вы должны написать сортировку в соответствии с вашим вариантом.

Формат входного файла

В первой строке входного файла содержится число n ($1 \leq n \leq 100000$) — количество элементов в массиве. Во второй строке находятся n целых чисел, по модулю не превосходящих 10^9 .

Формат выходного файла

В выходной файл надо вывести этот же массив в порядке неубывания, между любыми двумя числами должен стоять ровно один пробел.

Пример

<code>sort.in</code>	<code>sort.out</code>
10 1 8 2 1 4 7 3 2 3 6	1 1 2 2 3 3 4 6 7 8

Задача В. Двоичный поиск

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

Дан массив из n элементов, упорядоченный в порядке неубывания и m запросов: найти первое и последнее вхождение числа в массив.

Формат входного файла

В первую строке входного файла содержится одно число n — размер массива. ($1 \leq n \leq 100000$). Во второй строке находится n чисел в порядке неубывания — элементы массива. В третьей строке находится число m — количество запросов. В следующей строке находится m чисел — запросы.

Формат выходного файла

Для каждого запроса выведите в отдельной строке номер первого и последнего вхождения этого числа в массив. Если числа в массиве нет выведите два раза -1 .

Пример

<code>binsearch.in</code>	<code>binsearch.out</code>
5	1 2
1 1 2 2 2	3 5
3	-1 -1
1 2 3	

Задача С. К-ая порядковая статистика

Имя входного файла: kth.in

Имя выходного файла: kth.out

Дан массив из n элементов. Какое число k -ое в порядке возрастания в этом массиве.

Формат входного файла

В первую строке входного файла содержится два числа n — размер массива и k ($1 \leq k \leq n \leq 3 \cdot 10^7$). Во второй строке находятся числа A, B, C, a_1, a_2 по модулю не превосходящие 10^9 . Вы должны получить элементы массива начиная с третьего по формуле: $a_i = A \cdot a_{i-2} + B \cdot a_{i-1} + C$. Все вычисления должны производиться в 32 битном знаковом типе, переполнения должны игнорироваться.

Формат выходного файла

Выведите значение k -ое в порядке возрастания число в массиве a .

Пример

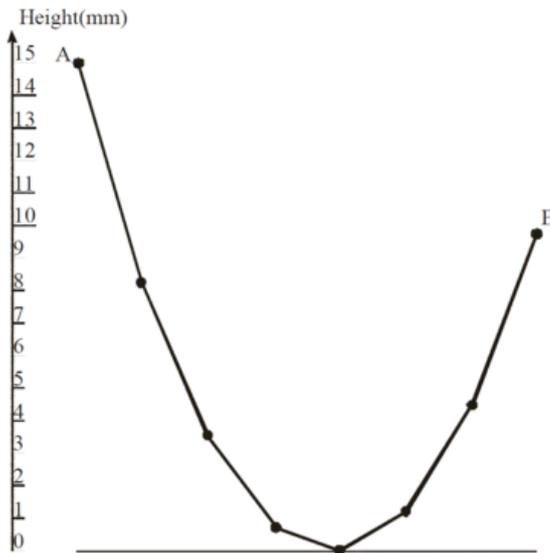
kth.in	kth.out
5 3 2 3 5 1 2	13
5 3 200000 300000 5 1 2	2

Во втором примере элементы массива a равны: (1, 2, 800005, -516268571, 1331571109).

Задача D. Гирлянда

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

Гирлянда состоит из n лампочек на общем проводе. Один её конец закреплён на заданной высоте A мм ($h_1 = A$). Благодаря силе тяжести гирлянда прогибается: высота каждой неконцевой лампы на 1 мм меньше, чем средняя высота ближайших соседей ($h_i = \frac{h_{i-1} + h_{i+1}}{2} - 1$ для $1 < i < N$). Требуется найти минимальную высоту второго конца B ($B = h_n$) при условии, что ни одна из лампочек не должна лежать на земле ($h_i > 0$ для $1 \leq i \leq N$).



Формат входного файла

В первую строке входного файла содержится два числа n и A ($3 \leq n \leq 1000$, n — целое, $10 \leq A \leq 1000$, A — вещественное).

Формат выходного файла

Вывести одно вещественное число B с двумя знаками после запятой.

Пример

<code>garland.in</code>	<code>garland.out</code>
8 15	9.75
692 532.81	446113.34

Задача Е. Цифровая сортировка

Имя входного файла: radixsort.in
Имя выходного файла: radixsort.out

Дано n строк, выведите их порядок после k фаз цифровой сортировки.

Формат входного файла

В первой строке входного файла содержится число n — количество строк, m — их длина и k — число фаз цифровой сортировки ($1 \leq n \leq 1000$, $1 \leq k \leq m \leq 1000$). В следующих n строках находятся сами строки.

Формат выходного файла

Выведите строки в порядке в котором они будут после k фаз цифровой сортировки.

Пример

radixsort.in	radixsort.out
3 3 1 bbb aba baa	aba baa bbb
3 3 2 bbb aba baa	baa aba bbb
3 3 3 bbb aba baa	aba baa bbb

Задача F. Анти-QuickSort

Имя входного файла: antiqs.in
Имя выходного файла: antiqs.out

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 : integer;
  key : integer;
  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

antigs.in	antigs.out
3	1 3 2

Задача G. Проверьте сортирующую сеть

Имя входного файла: `netcheck.in`

Имя выходного файла: `netcheck.out`

Проверьте является ли сеть из n проводов сортирующей.

Формат входного файла

В первой строке входного файла содержится три числа n — количество проводов, m — количество компараторов в сети и k — количество слоев в сети ($1 \leq n \leq 15$, $0 \leq m, k \leq 150$). В каждой из следующих строк содержится описание слоя из компараторов: число r — количество компараторов в слое и далее r пар чисел, номера проводов, которые сравнивает компаратор. Внутри слоя все номера проводов различны.

Формат выходного файла

Выведите «Yes», если сеть является сортирующей и «No», если нет.

Пример

<code>netcheck.in</code>	<code>netcheck.out</code>
4 6 3 2 1 2 3 4 2 1 4 2 3 2 1 2 3 4	Yes

Задача Н. Постройте сортирующую сеть

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

Постройте сортирующую сеть для n проводов.

Формат входного файла

В первой строке входного файла находится одно число n ($1 \leq n \leq 16$) — требуемый размер сортирующей сети.

Формат выходного файла

В первую строку выходного файла выведите три числа n — количество проводов, m — количество компараторов в сети и k — количество слоев в сети. В каждой из следующих строк выведите описание слоя из компараторов, число r — количество компараторов в слое и далее r пар чисел, номера проводов, которые сравнивает компаратор. Внутри слоя все номера проводов должны быть различны. Число слоев не должно превышать 12.

Пример

<code>netbuild.in</code>	<code>netbuild.out</code>
4	4 6 3 2 1 2 3 4 2 1 4 2 3 2 1 2 3 4