

Задача А. Дядюшка Ау и Сортировка

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

Ваш друг дядюшка Ау решил научиться программировать. Вы подарили ему книгу известного финского автора «Олимпиадное программирование». Дядюшка Ау за одну ночь прочитал подаренную книгу и бросился решать задачи.

Как и любой начинающий программист, дядюшка Ау часто ошибается по невнимательности. Противная тестирующая система не показывает, на каком тесте решение работает некорректно. Повезло, что вы уже опытный программист и можете помочь вашему другу с дебагом.

Обязательно ознакомьтесь с замечанием. Оно актуально для всех задач этого контекста.

Решение дядюшки Ау: <https://ejudge.lksh.ru/r/au-stress-lksh/sort.cpp>.

Формат входных данных

В первой строке дано целое число n ($1 \leq n \leq 10$) — Ау.

Во второй строке даны n целых чисел $a_1, a_2, \dots, a_n$ ($1 \leq a_i \leq 100$) — Ау.

Формат выходных данных

Выведите n целых чисел — Ау.

Замечание

В этом цикле задач вам предстоит сдавать в тестирующую систему тест, на котором программа дядюшки Ау работает неверно.

Вам известны ограничения на входные данные, но неизвестно условие задачи. Ограничения на входные данные умышленно занижены, чтобы найденный вами тест помог дядюшке дебагать. Попробуйте восстановить условие из решения дядюшки, написать **простое** корректное решение и стрессы.

Сдайте в тестирующую систему ломающий решение тест, удовлетворяющий формату входных данных.

Пример послышки:

```
3
3 2 1
```

Задача В. Дядюшка Ау и Сортировка 2

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

Вы помогли дядюшке Ау найти тест. Он потратил час на дебаг, но так и не справился найти багу. Помогите вашему другу исправить его решение, чтобы оно наконец прошло все тесты.

Обязательно ознакомьтесь с замечанием. Оно актуально для всех задач этого контекста.

Решение дядюшки Ау: <https://ejudge.lksh.ru/r/au-stress-lksh/sort.cpp>.

Пример

стандартный ввод	стандартный вывод
3 3 2 1	1 2 3

Замечание

В этом цикле задач вам предстоит сдавать в тестирующую систему исправленное решение дядюшки Ау.

Постарайтесь поменять как можно меньше строк, в противном случае вы получите вердикт «Нарушение правил оформления программ».

Задача C. Дядюшка Ау и Строка

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

Решение дядюшки Ау: <https://ejudge.lksh.ru/r/au-stress-lksh/hash.cpp>.

Формат входных данных

В первой строке дана строка из строчных латинских символов s ($1 \leq |s| \leq 20$) — Ау.

Формат выходных данных

Выведите единственное целое число — Ау.

Задача D. Дядюшка Ау и Строка 2

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

Вы помогли дядюшке Ау найти тест. Он потратил два часа на дебаг, но так и не справился найти багу. Помогите вашему другу исправить его решение, чтобы оно наконец прошло все тесты.

Решение дядюшки Ау: <https://ejudge.lksh.ru/r/au-stress-lksh/hash.cpp>.

Пример

стандартный ввод	стандартный вывод
abacaba	3

Задача E. Дядюшка Ау и Математика

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

Решение дядюшки Ау: <https://ejudge.lksh.ru/r/au-stress-lksh/math.cpp>.

Формат входных данных

В первой строке дано целое число n ($1 \leq n \leq 10$) — Ау.

Во второй строке даны n целых чисел $a_1, a_2, \dots, a_n$ ($1 \leq a_i \leq 2 \cdot 100$) — Ау.

Формат выходных данных

Выведите единственное целое число — Ау по модулю $10^9 + 7$.

Задача F. Дядюшка Ау и Математика 2

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

Вы помогли дядюшке Ау найти тест. Он потратил три часа на дебаг, но так и не справился найти багу. Помогите вашему другу исправить его решение, чтобы оно наконец прошло все тесты.

Решение дядюшки Ау: <https://ejudge.lksh.ru/r/au-stress-lksh/math.cpp>.

Пример

стандартный ввод	стандартный вывод
2 5 21	8

Задача G. Дядюшка Ау и Дерево отрезков

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

Решение дядюшки Ау: <https://ejudge.lksh.ru/r/au-stress-lksh/segtree.cpp>.

Формат входных данных

В первой строке даны целые числа n, q ($1 \leq n \leq 10, 1 \leq q \leq 15$) — Ау и Ау.

В следующих q строках даны Ау:

- $1 \ l \ r$ ($1 \leq l \leq r \leq n$) — Ау первого типа;
- $2 \ l \ r$ ($1 \leq l \leq r \leq n$) — Ау второго типа.

Формат выходных данных

Выведите в отдельной строке целое число для каждого Ау второго типа.

Задача Н. Дядюшка Ау и Дерево отрезков 2

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

Вы помогли дядюшке Ау найти тест. Он потратил четыре часа на дебаг, но так и не справился найти багу. Помогите вашему другу исправить его решение, чтобы оно наконец прошло все тесты.

Решение дядюшки Ау: <https://ejudge.lksh.ru/r/au-stress-lksh/segtree.cpp>.

Пример

стандартный ввод	стандартный вывод
3 4 1 1 2 1 3 3 1 1 1 2 2 3	2

Задача I. Серёжа и массив

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

Сегодня на доске в комповнике Серёжа увидел зацикленный массив чисел (т. е. после последнего элемента идёт первый). Серёжа заметил, что в массиве есть один локальный максимум (или, возможно, несколько равных подряд идущих максимальных элементов) и один локальный минимум (или несколько равных подряд идущих). Он понял, что в таком массиве можно быстро искать позиции максимума и минимума следующим образом:

1. Возьмём произвольный индекс i ;
2. Возьмём число k такое, что $2^k \leq n < 2^{k+1}$;
3. Совершаем k шагов. На каждом шаге рассматриваем элементы в позициях $i - 2^k, i, i + 2^k$, переходим в ту позицию, где элемент максимальный; при равенстве минимизируем индекс элемента.

Серёжа попытался применить этот алгоритм к массиву, написанному на доске, и понял, что алгоритм работает неверно. Найдите и вы такой тест, на котором алгоритм даст неправильный ответ!

Формат выходных данных

В этой задаче в тестирующую систему требуется сдать ровно один текстовый файл — тест, на котором описанное решение работает неверно. Если существует хоть одна начальная позиция, для которой алгоритм даёт неверный ответ, ваше решение будет засчитано как правильное.

Задача J. Серёжа и рюкзак

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

Сегодня Серёжа послушал лекцию Сергея Владимировича про задачу о рюкзаке.

Напомним формулировку задачи о рюкзаке:

Даны n предметов с положительными целыми весами w_i и стоимостями $cost_i$. И рюкзак размера W . Выберите, какие предметы положить в рюкзак, чтобы их суммарный вес не превосходил W , а суммарная стоимость была максимальна.

Серёжа придумал такой алгоритм решения этой задачи:

Переберём i — какой один предмет точно нужно положить в рюкзак. Оставшиеся предметы ($n - 1$ штук) отсортируем по убыванию удельной стоимости $\frac{cost_i}{w_i}$, а при равенстве удельной стоимости по возрастанию индекса (стабильная сортировка). Переберём оставшиеся предметы в таком порядке и будем пытаться жадно добавлять в рюкзак. Из полученных n ответов выберем лучший.

Он быстро понял, что этот алгоритм работает неверно на некоторых данных, но пока не смог найти контрпример. Помогите Серёже найти тест, на котором описанное решение даёт неправильный ответ!

Формат выходных данных

В этой задаче в тестирующую систему требуется сдать ровно один текстовый файл — тест, на котором описанное решение работает неверно.

Задача К. Два станка

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

Имеется множество из n работ и два станка. Время выполнения i -й работы на первом станке равно a_i , время выполнения i -й работы на втором станке равно b_i .

Каждую работу надо выполнить сначала на первом станке, потом на втором. На первом и на втором станках работы нужно выполнять в одинаковом порядке. Каждый станок в каждый момент времени может выполнять только одну работу.

Задача о двух станках — минимизировать время выполнения последней работы на втором станке.

Ваша задача — построить тесты к всевозможным жадностям.

Формат входных данных

В единственной строке написан тип жадности, к которой надо построить тест. Жадность — запуск `stable_sort` с компаратором, зависящим от типа жадности:

1. $a[i] < a[j]$
2. $b[i] < b[j]$
3. $a[i] + b[i] < a[j] + b[j]$
4. $a[i] - b[i] < a[j] - b[j]$
5. $a[i] * b[i] < a[j] * b[j]$
6. $(\text{double})a[i] / b[i] < (\text{double})a[j] / b[j]$
7. $\min(a[i], b[j]) < \min(a[j], b[i])$
8. $\max(a[i], b[j]) < \max(a[j], b[i])$
9. $\max(a[i], b[i]) < \max(a[j], b[j])$

Формат выходных данных

В первой строке выведите одно целое число n — длины массивов.

Во второй строке выведите n целых чисел — массив a ;

В третьей строке выведите n целых чисел — массив b .