

**Задача А. Транзисторы над Пекином**

Имя входного файла: transistor.in  
 Имя выходного файла: transistor.out  
 Ограничение по времени: 2 секунды  
 Ограничение по памяти: 64 мегабайта

Всемирно известный профессор В.В. Адимов продолжает свои разнообразные исследование устойчивости транзисторов. Теперь в голову ему пришла следующая задача: в доме  $N$  этажей, профессор хочет выяснить номер максимального этажа, падение с которого оставляет транзистор целым. Поскольку профессор исследует сферические транзисторы в вакууме, то можете считать что разбившись при падении с этажа  $f$  транзистор обязательно разобьется при падении с этажа  $f + 1$ . Дополнительно поставлено условие, что разрешено проведение не более чем  $K$  испытаний.

Эта задача была поручена именно вам, как самому успешному аспиранту профессора Адимова. Поскольку транзисторы нынче в цене, но наука все-таки дороже, то необходимо выяснить, какое минимальное количество транзисторов необходимо закупить, чтобы успешно провести эксперимент даже если вам будет катастрофически не везти.

**Формат входного файла**

В первой и единственной строке входного файла содержатся два целых положительных числа  $N$  и  $K$  не превосходящих 2000.

**Формат выходного файла**

Выведите единственное число - ответ на поставленную задачу. Если для данных  $N$  и  $K$  возможна ситуация, при которой мы не сможем получить ответ на вопрос даже имея неограниченный запас бесплатных транзисторов выведите  $-1$ .

**Примеры**

transistor.in	transistor.out
4 2	-1
4 3	2

**Задача В. Классический рюкзак с восстановлением**

Имя входного файла: backpack.in  
 Имя выходного файла: backpack.out  
 Ограничение по времени: 2 секунды  
 Ограничение по памяти: 64 мегабайта

Дано  $n$  предметов массой  $m_0, \dots, m_{n-1}$  и стоимостью  $c_0, \dots, c_{n-1}$  соответственно.

Ими наполняют рюкзак, который выдерживает вес не более  $M$ . Какую наибольшую стоимость могут иметь предметы в рюкзаке?

**Формат входного файла**

В первой строке вводится натуральное число  $n$ , не превышающее 100, и натуральное число  $M$ , не превышающее 10 000.

Во второй строке вводятся  $n$  натуральных чисел  $m_i$ , не превышающих 100.

Во третьей строке вводятся  $n$  натуральных чисел  $c_i$ , не превышающих 100.

**Формат выходного файла**

В первой строке выходного файла выведите одно целое число: наибольшую возможную стоимость рюкзака. Во второй строке выведите количество предметов, которое необходимо взять, чтобы набрать максимальную стоимость. В третьей строке выведите через пробел номера этих предметов. Нумерация предметов начинается с нуля. Если оптимальных наборов предметов несколько, выведите любой из них. Номера предметов можно выводить в произвольном порядке.

**Примеры**

backpack.in	backpack.out
4 6	13
2 4 1 2	3
7 2 5 1	0 2 3

**Задача С. Собеседование**

Имя входного файла: interview.in  
 Имя выходного файла: interview.out  
 Ограничение по времени: 2 секунды  
 Ограничение по памяти: 64 мегабайта

На собеседовании при поступлении на работу в компанию «Sasart» кандидатам предлагается решить задачу о наибольшей общей возрастающей подпоследовательности двух последовательностей чисел.

Суть задачи сводится к следующему: из последовательности чисел  $\{a_i\}$  необходимо выделить подпоследовательность  $\{a_{i_k}\}$  такую, что она:

- возрастает, т. е.  $\forall k a_{i_k} < a_{i_{k+1}}$ ;
- является подпоследовательностью последовательности  $b_i$ ;
- имеет длину, не меньшую чем все последовательности, обладающие предыдущими двумя свойствами.

Вам же для проверки правильности решения кандидатами этой задачи необходимо научиться вычислять хотя бы длину такой подпоследовательности.

**Формат входного файла**

В первой строке входного файла даны два целых числа  $n$  и  $m$  ( $1 \leq n, m \leq 5000$ ) — длины последовательностей  $\{a_i\}$  и  $\{b_i\}$ . Вторая и третья строки содержат, соответственно, по  $n$  и  $m$  натуральных чисел, не превосходящих 10 000 — сами последовательности.

**Формат выходного файла**

В выходной файл выведите единственное целое число — длину последовательности, обладающей описанными свойствами.

### Примеры

interview.in	interview.out
6 5 2 3 1 4 6 5 1 2 5 4 6	3

### Задача D. Дубы

Имя входного файла: `oaks.in`  
Имя выходного файла: `oaks.out`  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 64 мегабайта

На аллее перед зданием Министерства Обороны в ряд высажены  $n$  дубов. В связи с грядущим приездом главнокомандующего, было принято решение срубить несколько деревьев для придания аллее более милитаристического вида.

Внутренние распорядки министерства позволяют срубить дуб только в двух случаях:

- если и ближайший дуб слева, и ближайший дуб справа строго ниже, чем данный дуб;
- если и ближайший дуб слева, и ближайший дуб справа строго выше, чем данный дуб.

В частности, согласно этому правилу, нельзя срубить крайний левый и крайний правый дубы.

Министр хочет выработать такой план вырубki, чтобы в итоге осталось несколько дубов, высоты которых образуют неубывающую последовательность, то есть чтобы каждый дуб был не ниже, чем все дубы, стоящие слева от него. При этом, как человек любящий флору, министр хочет, чтобы было срублено минимальное возможное количество деревьев.

Помогите сотрудникам министерства составить оптимальный план вырубki аллеи или выяснить, что срубить дубы соответствующим образом невозможно.

### Формат входного файла

Первая строка входного файла содержит целое число  $n$  — количество дубов, растущих на аллее ( $2 \leq n \leq 200$ ). Вторая строка содержит  $n$  чисел — высоты дубов, приведённые слева направо. Высоты дубов — положительные целые числа, не превышающие 1000.

### Формат выходного файла

Если оставить последовательность дубов с неубывающими высотами невозможно, выходной файл должен содержать только одно число  $-1$ .

В случае, если искомый план существует, в первую строку выходного файла выведите целое число  $m$  — минимальное количество дубов, которые необходимо срубить. В следующие  $m$  строк выведите оптимальный план вырубki деревьев — номера дубов в том порядке, в котором их следует срубить, по одному номеру на строке.

Дубы нумеруются слева направо натуральными числами от 1 до  $n$ .

Если планов с наименьшим числом срубаемых дубов несколько, выведите любой из них.

### Примеры

oaks.in	oaks.out
5 3 2 4 8 5	2 2 4

### Задача E. Покраска забора

Имя входного файла: `paint.in`  
Имя выходного файла: `paint.out`  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 64 мегабайта

Мэр города Многоярославца решил построить перед своим домом забор из  $n$  деревянных досок и нанять лучшего маляра города для его покраски. Поскольку забор должен стать главной достопримечательностью города, лучший дизайнер города для каждой доски назначил тщательно выбранный цвет, в который она должна быть покрашена.

Для покраски главный маляр решил применить новейшую технологию, специально разработанную им для выполнения этого задания. Покраской забора будет заниматься специальный робот, который за один час может покрасить произвольный отрезок забора (набор соседних досок) в некоторый цвет. Поскольку задание должно быть выполнено как можно быстрее, требуется составить программу для робота, которая позволит достичь требуемой раскраски за минимальное время. Оставить какую-то из досок непокрашенной, естественно, запрещается.

### Формат входного файла

В первой строке входного файла записано число  $n$  ( $1 \leq n \leq 300$ ), где  $n$  количество досок в заборе. Вторая строка содержит строку из  $n$  символов, описывающую требуемую покраску забора. Цвета обозначаются заглавными латинскими буквами.

### Формат выходного файла

В первой строке выходного файла выведите  $m$  - наименьшее возможное время покраски забора в часах. Следующие  $m$  строк должны содержать программу покраски для робота. Каждая строка должна содержать два числа  $l_i$  и  $r_i$ , а также заглавную букву латинского алфавита, задающую цвет  $c_i$  и означает, что робот должен покрасить участок забора с  $l_i$  по  $r_i$ - доску в цвет  $c_i$  (если длина забора  $n$ , должно выполняться  $1 \leq l_i \leq r_i \leq n$ ).

### Примеры

paint.in	paint.out
5 ABBCA	3 1 5 A 4 4 C 2 3 B
2 AA	1 1 2 A