

**Задача А. Числа (12,5 баллов)**

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

Дана последовательность чисел  $a_1, a_2, \dots, a_n$ . За одну операцию разрешается удалить любое (кроме крайних) число, заплатив за это штраф, равный произведению этого числа на сумму соседних. Требуется удалить все числа, кроме крайних, с минимальным суммарным штрафом.

Пример начальной последовательности:

**1 50 51 50 1**

удаляем четвертое число, штраф  $50 \cdot (1 + 51) = 2600$ , получаем

**1 50 51 1**

удаляем третье число, штраф  $51 \cdot (50 + 1) = 2601$ , получаем

**1 50 1**

удаляем второе число, штраф  $50 \cdot (1 + 1) = 100$ .

Итого, штраф 5301.

**Формат входного файла**

В первой строке входного файла расположено одно число  $n$  ( $1 \leq n \leq 100$ ) — количество чисел в последовательности.

Во второй строке находятся  $n$  целых чисел  $a_1, a_2, \dots, a_n$ ; никакое из чисел по модулю не превосходит 100.

**Формат выходного файла**

Выведите в выходной файл одно число — минимальный суммарный штраф.

**Примеры**

<code>numbers.in</code>	<code>numbers.out</code>
5 1 50 51 50 1	5301

**Задача В. Деловые встречи (12,5 баллов)**

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

Алексей — успешный предприниматель, и в течение одного дня у него бывает много встреч с разными деловыми партнёрами. К сожалению, встречи бывают разные и не все приносят ему радость, после других же настроение улучшается. Также, на многие встречи не стоит приходить в слишком плохом или хорошем настроении — результат таких встреч может быть не таким, какой хочется Алексею.

К счастью, недавно Алексей научился оценивать своё настроение с помощью целых чисел. После этого для каждой встречи он оценил, при каком максимальном и минимальном настроении стоит на неё приходить, а также как изменится его настроение после этой встречи. Теперь он хочет распланировать порядок встреч так, чтобы в течение дня совершить максимальное число встреч.

Ваша задача — написать программу, которая по информации о всех встречах и настроении Алексея в начале дня находит порядок проведения встреч такой, что их количество при этом максимально.

**Формат входного файла**

Первая строка входного файла содержит два целых числа  $n$  и  $k$  ( $1 \leq n \leq 20$ ,  $-100 \leq k \leq 100$ ) — количество встреч и настроение Алексея в начале дня.

Следующие  $n$  строк содержат по три целых числа  $a_i, b_i$  и  $c_i$  ( $-100 \leq a_i, b_i, c_i \leq 100$ ) — минимальное и максимальное настроение, при котором встреча возможна, и изменение настроения по окончании встречи, соответственно.

**Формат выходного файла**

В первой строке выходного файла выведите число  $m$  — максимально возможно число встреч. В следующей строке выведите  $m$  целых чисел — номера встреч в порядке их проведения. Встречи пронумерованы в порядке описания во входном файле.

Если ответов с максимальным числом встреч несколько, выведите любой.

**Примеры**

<code>meetings.in</code>	<code>meetings.out</code>
3 0 1 3 3 0 1 2 1 3 1	3 2 3 1
3 1 -10 -5 3 -5 5 -2 -3 2 1	2 2 3

**Задача С. Путешествия в реальности (12,5 баллов)**

Имя входного файла: `alternative.in`  
 Имя выходного файла: `alternative.out`  
 Ограничение по времени: 1 секунда  
 Ограничение по памяти: 64 мегабайта

Каждый раз, когда в мире происходит значимое событие, наша реальность разветвляется на несколько — в зависимости от исхода этого события. После этого существует уже не только наша основная реальность, но и ответвившиеся от неё в моменты появления разных исходов.

Однажды один архимаг решил сделать мир лучше. Такая грандиозная задача не под силу одному архимагу, поэтому он решил найти самого себя ещё в  $K$  реальностях и выполнить эту задачу вместе. Проведённое теоретическое исследование показало, что, кроме реальности, в которой находится именно он, существует ещё  $N - 1$  реальностей. Для удобства они были занумерованы числами от 1 до  $N$ , при этом его собственная реальность имеет номер 1, а посетить ему необходимо реальности с номерами  $2, 3, \dots, K + 1$ .

Как уже говорилось, каждая реальность когда-то ответвилась от некоторой другой, за исключением одной Начальной реальности, которая существовала всегда (её номер может оказаться каким угодно; считается, что она появилась в момент времени 0). Исследование показало, что реальность с номером  $i$  ответвилась от реальности с номером  $P_i$  в момент времени  $T_i$ . Из каждой реальности с номером  $i$  архимаг может переместиться

- в любую ответвившуюся от неё, то есть в любую  $j$ , такую что  $P_j = i$ ;
- в  $P_i$ , если  $i$  — не Начальная реальность.

Другими словами, возможны лишь переходы вида  $i \rightleftharpoons P_i$ . На каждый такой переход в любую сторону архимаг затрачивает  $T_i - TP_i > 0$  условных единиц энергии.

Требуется найти минимальное количество энергии, которое потребуется архимагу, чтобы, начав в реальности с номером 1, посетить все реальности с номерами от 2 до  $K + 1$  (в любом порядке) и затем вновь вернуться в 1. Любую реальность при этом разрешается посещать сколько угодно раз.

### Формат входного файла

Сначала вводятся два целых числа  $N$  и  $K$  ( $0 \leq K < N \leq 100\,000$ ): количество доступных реальностей и количество реальностей, которые необходимо посетить. Далее идёт  $N$  пар целых чисел,  $i$ -я пара — это  $P_i$  и  $T_i$  ( $1 \leq P_i \leq N$ ,  $0 \leq T_i \leq 10^6$ ; для Начальной реальности  $P_i = T_i = 0$ ).

Гарантируется, что ответвившаяся реальность появилась строго позже породившей ( $T_i > TP_i$ ), и что маг может при желании добраться до любой из  $N$  реальностей.

### Формат выходного файла

Выведите единственное число  $E$  — минимальную возможную энергию, которая требуется архимагу для путешествия.

### Примеры

alternative.in	alternative.out
5 2	30
4 2	
4 6	
1 9	
0 0	
1 7	

### Задача D. Дубы (12,5 баллов)

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 распоряжки 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 выясните, 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 искомым 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	2
3 2 4 8 5	2
	4