

## Задача А. Слизни

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

$N$  слизней стоят в ряду.  $i$ -й слизень слева имеет размер  $a_i$ .

Мальчик Таро хочет объединить всех слизней в одного большого слизня. Он хочет этого добиться, используя следующую операцию, пока не останется один слизень: Таро выбирает двух соседних слизней и сливает их в одного. Новый слизень имеет размер  $x + y$ , если размеры двух слизней до слияния  $x$  и  $y$ . Стоимость этой операции —  $x + y$ . Общее расположение слизней после слияния двух не меняется.

Найдите минимальную стоимость, за которую можно объединить все  $N$  слизней в одного.

### Формат входных данных

Все входные данные — целые числа.

$$2 \leq N \leq 400$$

$$1 \leq a_i \leq 10^9$$

### Формат выходных данных

Выведите минимальную стоимость, за которую можно слить всех слизней в одного.

### Примеры

стандартный ввод	стандартный вывод
4 10 20 30 40	190
5 10 10 10 10 10	120

## Задача В. 17 стульев

Имя входного файла: стандартный ввод  
Имя выходного файла: стандартный вывод  
Ограничение по времени: 1 секунда  
Ограничение по памяти: 64 мегабайта

64 мегабайта

Остап Бендер снова пытается получить причитающиеся драгоценности, но на этот раз они были заперты в шкатулке, для открытия которой необходимо иметь  $N$  ключей. По закономерной случайности каждый из ключей был спрятан в одном из  $N$  стульев, распроданных на недавнем аукционе. После аукциона эти стулья были развезены в  $N$  городов.

И вот теперь Остап решил на новую безумную затею: заехать в каждый из городов и, провернув в каждом из них аферу, выкрасть необходимые ключи. Чтобы избежать конфликтов с недоброжелателями, Остап не хочет больше одного раза появляться в каком-либо городе. Также у Остапа есть список цен за проезд между каждой парой городов. Изначально Остап находится в городе под номером 1 и после посещения всех городов может незаметно скрыться из этой страны.

Помогите Остапу найти порядок посещения городов, при котором ему потребуется потратить как можно меньше средств на странствия, и тогда, возможно, он поделится с Вами добытыми бриллиантами.

### Формат входных данных

Первая строка содержит единственное число  $N$  — количество городов ( $1 \leq N \leq 17$ ).

Следующие  $N$  строк содержат по  $N$  целых неотрицательных чисел.  $j$ -тое число в  $i$ -й строке означает стоимость проезда из города  $i$  в город  $j$  ( $0 \leq a_{ij} \leq 100$ ). Если  $a_{ij} > 0$ , то проезд стоит  $a_{ij}$  рублей, иначе — это означает, что из города  $i$  в  $j$  невозможно проехать напрямую.

### Формат выходных данных

В первой строке выведите минимальную сумму денег, необходимую для посещения всех городов Остапом. В следующей строке выведите  $N$  чисел — порядок посещения городов, при котором эта сумма достигается. Если затею Остапа невозможно вывести, то в единственной строке выходного файла выведите число -1.

### Примеры

стандартный ввод	стандартный вывод
3 0 3 2 3 0 6 2 6 0	8 1 3 2
5 0 6 4 0 0 6 0 7 0 7 4 7 0 0 0 0 0 0 0 2 0 7 0 2 0	20 1 3 2 5 4

## Задача С. Распил брусьев

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

Вам нужно распилить деревянный брус на несколько кусков в заданных местах. Распилочная компания берет  $k$  рублей за распил одного бруска длиной  $k$  метров на две части.

Понятно, что различные способы распила приводят к различной суммарной стоимости заказа. Например, рассмотрим брус длиной 10 метров, который нужно распилить на расстоянии 2, 4 и 7 м, считая от одного конца. Это можно сделать несколькими способами. Можно распилить сначала на отметке 2 м, потом 4 и, наконец, 7 м. Это приведет к стоимости  $10+8+6=24$ , потому что сначала длина бруса, который пилили, была 10 м, затем она стала 8 м, и, наконец, 6 м. А можно распилить иначе: сначала на отметке 4 м, затем 2, затем 7 м. Это приведет к стоимости  $10+4+6=20$ , что лучше.

Определите минимальную стоимость распила бруса на заданные части.

### Формат входных данных

Первая строка входных данных содержит целое число  $L$  ( $2 \leq L \leq 10^6$ ) - длину бруса и целое число  $N$  ( $1 \leq N \leq 100$ ) - количество распилов. Во второй строке записано  $N$  целых чисел  $C_i$  ( $0 < C_i < L$ ) в строго возрастающем порядке - места, в которых нужно сделать распилы.

### Формат выходных данных

Выведите одно натуральное число - минимальную стоимость распила.

### Пример

стандартный ввод	стандартный вывод
10 3 2 4 7	20

## Задача D. Восточные забавы

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

Си Люнь Шань раскладывает рисовые шарики разного размера в ряд. Он хочет слепить как можно более крупные рисовые шарики, чтобы его друг мог их съесть. Си Люнь Шань может выполнять следующие действия:

1) Если два соседних рисовых шарика имеют одинаковый размер, Си Люнь Шань может объединить их, чтобы сделать новый рисовый шарик. Размер нового рисового шарика равен сумме размеров двух старых рисовых шариков. Он занимает место в ряду, которое ранее занимали два старых рисовых шарика.

2) Если два рисовых шарика одинакового размера и между ними ровно один рисовый шарик, Си Люнь Шань может объединить все три рисовых шарика, чтобы сделать новый рисовый шарик (средний рисовый шарик необязательно должен быть того же размера, что и два других). Размер нового рисового шарика равен сумме размеров трех старых рисовых шариков. Он занимает позицию в ряду, ранее занимаемую тремя старыми рисовыми шариками.

Си Люнь Шань может выполнять каждую операцию столько раз, сколько захочет.

Определите размер самого большого рисового шарика в ряду после выполнения 0 или более операций.

### Формат входных данных

В первой строке будет указано целое число  $N$  ( $1 \leq N \leq 400$ ) — количество шариков.

В следующей строке через пробел будут указаны  $N$  целых чисел  $a_i$  ( $1 \leq a_i \leq 10^6$ ), представляющие размеры рисовых шариков в порядке слева направо.

### Формат выходных данных

Выведите размер самого большого рисового шарика, который может сформировать Си Люнь Шань.

### Система оценки

Тестирование происходит по группам, баллы за группы суммируются, но начисляются только при прохождении всех тестов группы:

- 1) Тесты 1-2 из условия и стоят 0 баллов
- 2) Тесты 3-10 имеют ограничение  $N \leq 4$  и стоят 5 баллов
- 3) Тесты 11-22 имеют ограничение  $N \leq 10$  и стоят 15 баллов
- 4) Тесты 23-33 имеют ограничение  $N \leq 50$  и стоят 30 баллов
- 5) Оставшиеся имеют ограничение  $N \leq 400$  и стоят 50 баллов

### Примеры

стандартный ввод	стандартный вывод
7 47 12 12 3 9 9 3	48
4 1 2 3 1	3

### Замечание

\*\*Пояснение к выводам для примера 1\*\*

Один из возможных способов получения рисовых шариков 48-го размера — это объединить 12 и 12, чтобы получился рисовый шарик 24-го размера. Затем объедините 9 и 9, чтобы получился рисовый шарик 18-го размера. Затем соедините 3, 18 и 3, чтобы получился рисовый шарик 24-го размера. Наконец, соедините два рисовых шарика 24-го размера, чтобы получился рисовый шарик 48-го размера.

\*\*Пояснение к выводам для примера ввода 2\*\*

Ходы делать не получится, поэтому самый большой рисовый шарик в ряду имеет размер 3.

## Задача Е. Справедливый дележ

Имя входного файла: стандартный ввод  
Имя выходного файла: стандартный вывод  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 256 мегабайт

— Я хотел честно, — сказал Балаганов, собирая деньги с кровати, — по справедливости.

В коробке от сигарет «Кавказ», отнятой у Корейко, было всего  $1 \leq N \leq 15$  купюр, каждая номиналом  $1 \leq a_i \leq 10^3$ . Разделить надо было на  $1 \leq K \leq 100$  частей, причём известно, что общая сумма делится на  $K$ . Метод деления «по справедливости» следующий: если разделить поровну не получается, то делить следует так, чтобы среднеквадратичное отклонение было минимальным.

Среднеквадратичное отклонение для данного способа дележа определяется следующим образом. Пусть  $i$ -й человек получил сумму денег, равную  $S_i$ . Обозначим за  $M$  среднее арифметическое сумм денег, полученных каждым:  $M = (\sum_{i=1}^K S_i)/K$ . Тогда среднеквадратичное отклонение можно вычислить так:  $\sigma = \sqrt{(\sum_{i=1}^K (S_i - M)^2)/K}$ .

— И как? — поинтересовался Остап.

— Сложно... — вздохнул Шура.

Попробуйте и вы разделить деньги «по справедливости».

### Формат входных данных

В первой строке входного файла задано  $N$  — количество купюр и  $K$  — количество участников дележа. Далее, в следующей строке, заданы  $N$  целых чисел  $a_i$  — номиналы купюр.

### Формат выходных данных

В первой строке выведите искомое среднеквадратичное отклонение с точностью до шести знаков после десятичной точки, в следующей строке выведите  $N$  чисел от 1 до  $K$  — номер участника дележа, которому досталась  $i$ -я купюра. Если оптимальных решений несколько, разрешается выводить любое.

### Примеры

стандартный ввод	стандартный вывод
4 3 1 2 3 6	1.41421356237309514547 1 1 2 3
1 1 179	0.00000000000000000000 1

## Задача F. Дубы

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

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 вырубки, 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 вырубки аллеи или выяснить, что срубить дубы соответствующим образом невозможно.

### Формат входных данных

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, приведенные слева направо. Высоты дубов — положительные целые числа, не превышающие 1 000.

### Формат выходных данных

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 вырубки 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.

### Пример

стандартный ввод	стандартный вывод
5	2
3 2 4 8 5	4
	2

## Задача G. Группировка кроликов

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

$N$  кроликов пронумерованы  $1, 2, \dots, N$ .

Для каждой пары  $i, j (1 \leq i, j \leq N)$  совместимость кроликов  $i, j$  описывается целым числом  $a_{i,j}$ .  
 $a_{i,i} = 0$  для каждого  $i (1 \leq i \leq N)$ , и  $a_{i,j} = a_{j,i}$  для каждой пары  $i, j (1 \leq i, j \leq N)$

Мальчик Таро хочет разделить  $N$  кроликов на какое-то количество групп. Один кролик должен принадлежать только одной группе.

После группировки, для каждой пары  $i, j (1 \leq i, j \leq N)$ , Таро получает  $a_{i,j}$  очков, если кролики  $i, j$  в одной группе.

Найдите максимальное количество очков, которое может набрать Таро.

### Формат входных данных

Все входные данные — целые числа.

$$1 \leq N \leq 16$$

$$|a_{i,j}| \leq 10^9$$

$$a_{i,i} = 0$$

$$a_{i,j} = a_{j,i}$$

### Формат выходных данных

Выведите максимальное количество очков, которое может набрать Таро.

### Примеры

стандартный ввод	стандартный вывод
3 0 10 20 10 0 -100 20 -100 0	20
2 0 -10 -10 0	0



## Задача Н. Разбиение на пути

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

Дан ориентированный граф. Найти число способов разбить все его вершины на простые пути. Каждая вершина должна лежать ровно в одном пути, каждый путь содержит не менее двух вершин. Разбиения на пути различны, если различны множества использованных рёбер.

### Формат входных данных

В первой строке входного файла находятся два натуральных числа  $N$  и  $M$  ( $1 \leq N \leq 17$ ,  $M \leq n(n - 1)$ ) — количество вершин и рёбер в графе соответственно. Далее в  $M$  строках перечислены рёбра графа. Каждое ребро задаётся парой чисел  $a, b$  — номерами начальной и конечной вершин соответственно. Рёбра не повторяются, у каждого ребра  $a \neq b$ .

### Формат выходных данных

Выведите число разбиений вершина графа на пути.

### Примеры

<code>vertex-partition.in</code>	<code>vertex-partition.out</code>
4 3 1 2 2 3 3 4	2
4 6 1 2 2 3 3 4 4 1 3 1 2 4	8

## Задача I. Деловые встречи

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
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 входных данных

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 выходных данных

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

стандартный ввод	стандартный вывод
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 3 2

## Задача J. Казино

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

Вновь открытое казино предложило оригинальную игру. В начале игры крупье выставляет в ряд несколько фишек разных цветов. Кроме того, он объявляет, какие последовательности фишек игрок может забирать себе в процессе игры. Далее игрок забирает себе одну из заранее объявленных последовательностей фишек, расположенных подряд. После этого крупье сдвигает оставшиеся фишки, убирая разрыв. Затем игрок снова забирает себе одну из объявленных последовательностей и так далее. Игра продолжается до тех пор, пока игрок может забирать фишки. Рассмотрим пример. Пусть на столе выставлен ряд фишек `rrrgggbbb`, и крупье объявил последовательности `rg` и `gb`. Игрок, например, может забрать фишки `rg`, лежащие на третьем и четвёртом местах слева. После этого крупье сдвинет фишки, и на столе получится ряд `rrggbbb`. Ещё дважды забрав фишки `rg`, игрок добьётся того, что на столе останутся фишки `bbb` и игра закончится, так как игроку больше нечего забрать со стола. Игрок мог бы действовать и по-другому - на втором и третьем ходах забрать не последовательности `rg`, а последовательности `gb`. Тогда на столе остались бы фишки `rrb`. Аналогично, игрок мог бы добиться того, чтобы в конце остались ряды `rrr` или `rbb`. После окончания игры полученные фишки игрок меняет на деньги. Цена фишки зависит от её цвета. Требуется написать программу, определяющую максимальную сумму, которую сможет получить игрок.

### Формат входных данных

В первой строке входных данных содержится число  $K$  ( $1 \leq K \leq 26$ ) - количество цветов фишек. Каждая из следующих  $K$  строк начинается со строчной латинской буквы, обозначающей цвет. Далее в той же строке через пробел следует целое число  $X_i$  ( $1 \leq X_i \leq 150, i = 1..K$ ) - цена фишки соответствующего цвета. В  $(K + 2)$ -ой строке описан ряд фишек, лежащих на столе в начале игры. Ряд задается  $L$  строчными латинскими буквами ( $1 \leq L \leq 150$ ), которые обозначают цвета фишек ряда. В следующей строке содержится число  $N$  ( $1 \leq N \leq 150$ ) - количество последовательностей, которые были объявлены крупье. В следующих  $N$  строках записаны эти последовательности. Гарантируется, что сумма длин этих  $N$  строк не превосходит 150 символов, и все они непустые.

### Формат выходных данных

Выведите единственное целое число — максимальную сумму денег, которую может получить игрок.

### Пример

стандартный ввод	стандартный вывод
3 v 3 l 1 u 2 luvu 3 luv vul uuu	6