

Задача 1. Сокращаем переменные

Имя входного файла: `pause.in`
Имя выходного файла: `pause.out`
Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайт

Требуется подсчитать, на сколько минут раньше будет заканчиваться k -й урок, если все переменные сократить на m минут.

Формат входного файла

Вводится два натуральных числа: k и m ($1 \leq k \leq 10^4$, $0 \leq m \leq 10^4$).

Формат выходного файла

Выведите одно натуральное число — время в минутах.

Пример входных и выходных файлов

<code>pause.in</code>	<code>pause.out</code>
3 5	10

Задача 2. Урок физкультуры

Имя входного файла: gym.in
Имя выходного файла: gym.out
Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайт

На уроке физкультуры тренер Андрей Сергеевич выстраивает учеников в одну шеренгу. В шеренге сначала идут мальчики, а потом девочки. При этом мальчики в шеренге стоят по убыванию роста, аналогично девочки тоже стоят по убыванию роста. Таким образом, следом за самым низким мальчиком стоит самая высокая девочка.

Андрея Сергеевича заинтересовал вопрос, какое максимальное различие в росте двух стоящих рядом учеников. Напишите программу, которая поможет Андрею Сергеевичу ответить на этот важный для него вопрос.

Формат входного файла

Первая строка содержит целое число n – число учеников в классе ($2 \leq n \leq 50$).

Следующие n строк содержат по два целых числа каждая: a_i и h_i – пол и рост в сантиметрах i -го ученика (a_i равно 0 или 1, $100 \leq h_i \leq 200$). Значение $a_i=0$ означает, что i -й ученик – мальчик, а значение $a_i=1$ означает, что i -й ученик – девочка.

Формат выходного файла

Выведите одно число – максимальное различие в росте стоящих рядом учеников после того, как они выстроятся в шеренгу на уроке физкультуры.

Пример входных и выходных файлов

gym.in	gym.out
6 0 120 1 130 1 142 1 115 0 145 0 134	22

Задача 3. Пятница, 13-е

Имя входного файла: `friday.in`
Имя выходного файла: `friday.out`
Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайт

Календарь жителей планеты Плюк состоит из N месяцев, каждый месяц состоит ровно из 30 дней, неделя состоит из 7 дней. Особо несчастливыми на планете Плюк считается 13-е число месяца, если оно выпадает на пятницу.

Известно, что Новый год на планете Плюк начался в k -й по счету день недели (1-й день недели – понедельник, 2-й – вторник, 3-й – среда, ..., 7-й – воскресенье).

Определите, сколько в этом году на планете Плюк будет особо несчастливых пятниц, 13-е.

Формат входного файла

Программа получает на вход два натуральных числа, записанных в отдельных строках. Первое число – количество месяцев в календаре планеты Плюк N , не превосходящее 10^9 . Второе число – номер дня недели, на который приходится первое число первого месяца нового года, может принимать значения от 1 до 7.

Формат выходного файла

Программа должна вывести единственное целое число – количество несчастливых дней в этом году.

Пример входных и выходных файлов

<code>friday.in</code>	<code>friday.out</code>
12 1	2

Пояснение к примеру

На 13-е число будут приходиться пятницы четвертого и одиннадцатого месяцев.

Задача 4. Сумма минимумов

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

У Саши есть блокнот, состоящий из n листочков, пронумерованных от 1 до n . На i -м листочке написано целое число a_i .

Аня собирается разорвать блокнот на k частей, для этого она выбирает $k-1$ число $1 \leq r_1 < r_2 < \dots < r_{k-1} < n$ и разрывает блокнот так, что листки с 1 по r_1 -й оказываются в первой части, листки с (r_1+1) -го по r_2 -й оказываются во второй части, и т.д., последняя k -я часть содержит листки с $(r_{k-1} + 1)$ -го по n -й.

После того, как Аня разорвет блокнот, Саша найдет минимальное число в каждой из получившихся частей и сложит их. Аня хочет разорвать блокнот таким образом, чтобы получившаяся сумма была как можно больше. Помогите ей выбрать способ разорвать блокнот, чтобы максимизировать сумму минимальных значений.

Формат входного файла

Первая строка ввода содержит два числа: n и k ($2 \leq k \leq n \leq 300$).

Вторая строка содержит n целых чисел: a_1, a_2, \dots, a_n ($1 \leq a_i \leq 10^9$).

Формат выходного файла

На первой строке выведите максимальное значение суммы, которое удастся достичь Ане.

На второй строке выведите значения r_1, r_2, \dots, r_{k-1} , которые ей необходимо выбрать. Если вариантов разорвать блокнот, чтобы максимизировать искомую сумму несколько, выведите любой из них.

Пример входных и выходных файлов

<code>summin.in</code>	<code>summin.out</code>
10 5	27
1 10 2 8 9 3 5 4 7 6	3 4 5 8

Пояснение к примеру

В приведенном примере Аня разорвала блокнот на части $[1, 10, 2]$, $[8]$, $[9]$, $[3, 5, 4]$ и $[7, 6]$.

Искомая сумма равна $1 + 8 + 9 + 3 + 6 = 27$.

Задача 1. Урок физкультуры

Имя входного файла: gym.in
Имя выходного файла: gym.out
Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайт

На уроке физкультуры тренер Андрей Сергеевич выстраивает учеников в одну шеренгу. В шеренге сначала идут мальчики, а потом девочки. При этом мальчики в шеренге стоят по убыванию роста, аналогично девочки тоже стоят по убыванию роста. Таким образом, следом за самым низким мальчиком стоит самая высокая девочка.

Андрея Сергеевича заинтересовал вопрос, какое максимальное различие в росте двух стоящих рядом учеников. Напишите программу, которая поможет Андрею Сергеевичу ответить на этот важный для него вопрос.

Формат входного файла

Первая строка содержит целое число n – число учеников в классе ($2 \leq n \leq 50$).

Следующие n строк содержат по два целых числа каждая: a_i и h_i – пол и рост в сантиметрах i -го ученика (a_i равно 0 или 1, $100 \leq h_i \leq 200$). Значение $a_i=0$ означает, что i -й ученик – мальчик, а значение $a_i=1$ означает, что i -й ученик – девочка.

Формат выходного файла

Выведите одно число – максимальное различие в росте стоящих рядом учеников после того, как они выстроятся в шеренгу на уроке физкультуры.

Пример входных и выходных файлов

gym.in	gym.out
6	22
0 120	
1 130	
1 142	
1 115	
0 145	
0 134	

Задача 2. Пятница, 13-е

Имя входного файла: `friday.in`
Имя выходного файла: `friday.out`
Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайт

Календарь жителей планеты Плюк состоит из N месяцев, каждый месяц состоит ровно из 30 дней, неделя состоит из 7 дней. Особо несчастливыми на планете Плюк считается 13-е число месяца, если оно выпадает на пятницу.

Известно, что Новый год на планете Плюк начался в k -й по счету день недели (1-й день недели – понедельник, 2-й – вторник, 3-й – среда, ..., 7-й – воскресенье).

Определите, сколько в этом году на планете Плюк будет особо несчастливых пятниц, 13-е.

Формат входного файла

Программа получает на вход два натуральных числа, записанных в отдельных строках. Первое число – количество месяцев в календаре планеты Плюк N , не превосходящее 10^9 . Второе число – номер дня недели, на который приходится первое число первого месяца нового года, может принимать значения от 1 до 7.

Формат выходного файла

Программа должна вывести единственное целое число – количество несчастливых дней в этом году.

Пример входных и выходных файлов

<code>friday.in</code>	<code>friday.out</code>
12	2
1	

Пояснение к примеру

На 13-е число будут приходиться пятницы четвертого и одиннадцатого месяцев.

Задача 3. Сумма минимумов

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

У Саши есть блокнот, состоящий из n листочков, пронумерованных от 1 до n . На i -м листочке написано целое число a_i .

Аня собирается разорвать блокнот на k частей, для этого она выбирает $k-1$ число $1 \leq r_1 < r_2 < \dots < r_{k-1} < n$ и разрывает блокнот так, что листки с 1 по r_1 -й оказываются в первой части, листки с (r_1+1) -го по r_2 -й оказываются во второй части, и т.д., последняя k -я часть содержит листки с $(r_{k-1} + 1)$ -го по n -й.

После того, как Аня разорвет блокнот, Саша найдет минимальное число в каждой из получившихся частей и сложит их. Аня хочет разорвать блокнот таким образом, чтобы получившаяся сумма была как можно больше. Помогите ей выбрать способ разорвать блокнот, чтобы максимизировать сумму минимальных значений.

Формат входного файла

Первая строка ввода содержит два числа: n и k ($2 \leq k \leq n \leq 300$).

Вторая строка содержит n целых чисел: a_1, a_2, \dots, a_n ($1 \leq a_i \leq 10^9$).

Формат выходного файла

На первой строке выведите максимальное значение суммы, которое удастся достичь Ане.

На второй строке выведите значения r_1, r_2, \dots, r_{k-1} , которые ей необходимо выбрать. Если вариантов разорвать блокнот, чтобы максимизировать искомую сумму несколько, выведите любой из них.

Пример входных и выходных файлов

<code>summin.in</code>	<code>summin.out</code>
10 5	27
1 10 2 8 9 3 5 4 7 6	3 4 5 8

Пояснение к примеру

В приведенном примере Аня разорвала блокнот на части $[1, 10, 2]$, $[8]$, $[9]$, $[3, 5, 4]$ и $[7, 6]$.

Искомая сумма равна $1 + 8 + 9 + 3 + 6 = 27$.

Задача 4. Урюк

Имя входного файла: apricot.in
Имя выходного файла: apricot.out
Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайт

В давние времена Золотая Орда ежегодно собирала дань золотыми монетами. Известный крымский хан Гирей решил схитрить: выплачивая дань из N золотых монет, он подложил среди них одну фальшивую – более легкую монету. Об этом донесли казначею Золотой Орды. Для обнаружения подделки он решил использовать магические весы, работающие на урюке.

На чаши магических весов кладутся две кучи монет. Весы устанавливаются, совпадает или различается вес этих куч. При этом, если кучи имеют разный вес, то весы указывают, какая из куч легче.

При совпадении веса обеих куч весы требуют R плодов урюка, а при несовпадении – U плодов. Казначей, сам любитель урюка, хочет и фальшивую монету обнаружить, и сэкономить на урюке.

Требуется написать программу, которая по заданному количеству монет N , при условии, что только одна из них легче других, укажет минимальное количество урюка, с помощью которого эта фальшивая монета гарантированно будет обнаружена.

Формат входного файла

Во входном файле в единственной строке находятся три целых числа N , R и U ($2 \leq N \leq 1\,000\,000$, $1 \leq R$, $U \leq 1\,000\,000$), где N – количество монет, R – количество плодов урюка, затрачиваемых в случае совпадения веса куч монет, U – количество плодов урюка, затрачиваемых в случае их различия. Все числа разделены пробелом.

Формат выходного файла

Выходной файл должен содержать одно число – минимальное количество урюка, с помощью которого гарантированно будет обнаружена фальшивая монета.

Пример входных и выходных файлов

apricot.in	apricot.out
4 3 1	2
3 3 1	3
15 2 3	8
10 2 1	3