

Задача А. Средняя зарплата

Имя входного файла: a.in
 Имя выходного файла: a.out
 Максимальное время работы на одном тесте: 1 секунда
 Максимальный объем используемой памяти: 64 мегабайта

В фирме MacroHard работают N сотрудников, каждый из которых получает зарплату, выражающуюся целым числом рублей. Известно, что ни один сотрудник не получает меньше 5000 рублей, и никто не получает больше 100000 рублей. Также известно, что средняя зарплата сотрудника в этой фирме выражается целым числом копеек и составляет A рублей B копеек.

Журналист, готовя публикацию об этой фирме, решил привести зарплаты всех сотрудников. Однако оказалось, что это коммерческая тайна. Журналиста это не смутило, и он решил придумать всем сотрудникам зарплаты. Однако у него возникла сложность – для правдоподобности должны выполняться все общеизвестные ограничения (зарплаты должны выражаться целым числом рублей из диапазона от 5000 до 100000, и вычисление средней зарплаты должно в точности приводить к результату A рублей B копеек).

Помогите ему! Напишите программу, которая по введенным числам N , A , B «придумает» и выведет N зарплат.

Формат входных данных

Вводятся натуральное число N ($1 \leq N \leq 100$), натуральное число A ($10000 \leq A \leq 30000$) и целое число B ($0 \leq B \leq 99$).

Формат выходных данных

Выведите N целых чисел, выражающих зарплаты сотрудников в рублях. Если возможных вариантов распределения зарплат несколько, выведите любой из них. Если распределить зарплаты с учетом наложенных условий невозможно, выведите одно число 0.

Примеры

a.in	a.out
5 10000 0	10000 10000 10000 10000 10000
2 20000 50	20000 20001

Задача В. Стройка

Имя входного файла: b.in
 Имя выходного файла: b.out
 Максимальное время работы на одном тесте: 1 секунда
 Максимальный объем используемой памяти: 64 мегабайта

На территории будущей стройки растут три дерева. Фирма получила разрешение на строительные работы с условием, что два (любых) дерева будут сохранены. Прораб хочет построить забор треугольной формы так, чтобы внутри него оказалось ровно два дерева.

Деревья на плане изображаются кругами, которые попарно не вложены друг в друга и не пересекаются (но могут касаться).

Напишите программу, которая по введенной информации о деревьях определит, возможно ли построить такой забор, и, если да, то какое дерево окажется не огорожено.

Формат входных данных

Вводится информация о трех деревьях: для каждого дерева координаты центра и радиус круга, изображающего это дерево на плане. Все числа целые, не превосходящие по модулю 3000. Радиус – натуральное число.

Формат выходных данных

Выведите одно число – номер дерева (деревья нумеруются начиная с 1 в порядке задания их во входных данных), которое окажется не огорожено. Если забор треугольной формы, огораживающий ровно два дерева, построить невозможно, выведите число 0. Если существует несколько решений, выведите любое.

Пример

b.in	b.out
0 0 1 3 0 1 6 0 1	3

Задача С. Кодовый замок

Имя входного файла: `c.in`
 Имя выходного файла: `c.out`
 Максимальное время работы на одном тесте: 1 секунда
 Максимальный объем используемой памяти: 64 мегабайта

Кодовый замок состоит из N рычажков, каждый из которых может быть установлен в любое из K положений, обозначенных натуральными числами от 1 до K . Известно, что для того чтобы открыть замок, нужно, чтобы сумма положений любых трех последовательных рычажков была равна K .

Два рычажка уже установлены в некоторые положения, и их переключать нельзя. Рычажок с номером p_1 установлен в положение v_1 , а рычажок p_2 — в положение v_2 .

Напишите программу, которая определит, сколькими способами можно установить остальные рычажки, чтобы открыть замок.

Формат входных данных

Вводятся натуральные числа N, K, p_1, v_1, p_2, v_2 .

$3 \leq N \leq 100\,000$, $3 \leq K \leq 100\,000$, $p_1 \neq p_2$, $1 \leq p_1 \leq N$, $1 \leq p_2 \leq N$, $1 \leq v_1 \leq K$, $1 \leq v_2 \leq K$.

Формат выходных данных

Выведите одно число — количество искомых комбинаций или 0, если, соблюдая все условия, замок открыть невозможно.

Примеры

<code>c.in</code>	<code>c.out</code>
3 3 1 1 2 1	1
3 3 1 1 3 2	0
4 4 1 1 4 1	2
5 3 1 1 4 1	1

Задача D. Смайлики

Имя входного файла: d.in
 Имя выходного файла: d.out
 Максимальное время работы на одном тесте: 1 секунда
 Максимальный объем используемой памяти: 64 мегабайта

Напишите программу, которая посчитает количество смайликов в заданном тексте.

Смайликом будем считать последовательность символов, удовлетворяющую условиям:

- первым символом является либо ; (точка с запятой) либо : (двоеточие) ровно один раз
- далее может идти символ - (минус) сколько угодно раз (в том числе символ минус может идти ноль раз)
- в конце обязательно идет некоторое количество (не меньше одной) одинаковых скобок из следующего набора: (,), [,].
- внутри смайлика не может встречаться никаких других символов.

Например, нижеприведенные последовательности являются смайликами:

```

        : )
    ; ----- [ [ [ [ [ [ [ [
    
```

в то время как эти последовательности смайликами не являются (хотя некоторые из них содержат смайлики):

```

        : - ) ]
        ; --
        - )
        : : - (
        : - ( )
    
```

В этой задаче надо будет посчитать количество смайликов, содержащихся в данном тексте.

Формат входных данных

Вводится одна строка текста, которая может содержать маленькие латинские буквы, пробелы, символы, которые могут встречаться в смайликах. Длина строки не превышает 100 000 символов.

Формат выходных данных

Выведите одно число — количество смайликов, которые встречаются в тексте.

Примеры

d.in	d.out
:) ; ----- [[[[[[[[2
: -)] ; ---- ;	1
-) (---- : ----	0
hello :-)	1

Задача Е. Субботник

Имя входного файла: e.in
 Имя выходного файла: e.out
 Максимальное время работы на одном тесте: 1 секунда
 Максимальный объем используемой памяти: 64 мегабайта

В классе учатся N человек. Классный руководитель получил указание направить на субботник R бригад по C человек в каждой.

Все бригады на субботнике будут заниматься переноской бревен. Каждое бревно одновременно несут все члены одной бригады. При этом бревно нести тем удобнее, чем менее различается рост членов этой бригады.

Числом неудобства бригады будем называть разность между ростом самого высокого и ростом самого низкого членов этой бригады (если в бригаде только один человек, то эта разница равна 0). Классный руководитель решил сформировать бригады так, чтобы максимальное из чисел неудобства сформированных бригад было минимально. Помогите ему в этом!

Рассмотрим следующий пример. Пусть в классе 8 человек, рост которых в сантиметрах равен 170, 205, 225, 190, 260, 130, 225, 160, и необходимо сформировать две бригады по три человека в каждой. Тогда одним из вариантов является такой:

1 бригада: люди с ростом 225, 205, 225

2 бригада: люди с ростом 160, 190, 170

При этом число неудобства первой бригады будет равно 20, а число неудобства второй — 30. Максимальное из чисел неудобств будет 30, и это будет наилучший возможный результат.

Формат входных данных

Сначала вводятся натуральные числа N , R и C — количество человек в классе, количество бригад и количество человек в каждой бригаде ($1 \leq R \cdot C \leq N \leq 100\,000$). Далее вводятся N целых чисел — рост каждого из N учеников. Рост ученика — натуральное число, не превышающее 1 000 000 000.

Формат выходных данных

Выведите одно число — наименьшее возможное значение максимального числа неудобства сформированных бригад.

Примеры

e.in	e.out
8 2 3 170 205 225 190 260 130 225 160	30

Задача F. Язык Мумба-Юмба

Имя входного файла: f.in
 Имя выходного файла: f.out
 Максимальное время работы на одном тесте: 1 секунда
 Максимальный объем используемой памяти: 64 мегабайта

Слова в языке Мумба-Юмба могут состоять только из букв **a**, **b** и при этом:

- никогда не содержат двух букв **b** подряд,
- ни в одном слове никогда не встречается три одинаковых подслова подряд. Например, по этому правилу в язык Мумба-Юмба не могут входить слова **aaa** (так как три раза подряд содержит подслово **a**), **ababab** (так как три раза подряд содержит подслово **ab**), **aabababa** (также три раза подряд содержит подслово **ab**).

Все слова, удовлетворяющие вышеописанным правилам, входят в язык Мумба-Юмба.

Напишите программу, которая подсчитает количество слов длины ровно K символов в языке племени Мумба-Юмба.

Формат входных данных

Вводится одно число K ($1 \leq K \leq 100\,000$)

Формат выходных данных

Выведите одно число — количество слов в этом языке длины K .

Примеры

f.in	f.out
1	2
2	3
3	4
5	7

Пояснение

Слова длины 1 — это слова **a**, **b**

Слова длины 2 — это слова **aa**, **ab**, **ba**

Слова длины 3 — это слова **aab**, **aba**, **baa**, **bab**

Слова длины 5 — это слова **aabaa**, **aabab**, **abaab**, **ababa**, **baaba**, **babaa**, **babab**

Задача G. Операционные системы

Имя входного файла: g.in
 Имя выходного файла: g.out
 Максимальное время работы на одном тесте: 1 секунда
 Максимальный объем используемой памяти: 64 мегабайта

Васин жесткий диск состоит из M секторов. Вася последовательно устанавливал на него различные операционные системы следующим методом: он создавал новый раздел диска из последовательных секторов, начиная с сектора номер a_i и до сектора b_i включительно, и устанавливал на него очередную систему. При этом если очередной раздел хотя бы по одному сектору пересекается с каким-то ранее созданным разделом, то ранее созданный раздел «затирается», и операционная система, которая на него была установлена, больше не может быть загружена.

Напишите программу, которая по информации о том, какие разделы на диске создавал Вася, определит, сколько в итоге работающих операционных систем установлено и в настоящий момент работает на Васином компьютере.

Формат входных данных

Сначала вводятся натуральное число M — количество секторов на жестком диске ($1 \leq M \leq 10^9$) и целое число N — количество разделов, которое последовательно создавал Вася ($0 \leq N \leq 100\,000$).

Далее идут N пар чисел a_i и b_i , задающих номера начального и конечного секторов раздела ($1 \leq a_i \leq b_i \leq M$).

Формат выходных данных

Выведите одно число — количество работающих операционных систем на Васином компьютере.

Примеры

g.in	g.out
10 3 1 3 4 7 3 4	1
10 4 1 3 4 5 7 8 4 6	3

Задача Н. Пицца

Имя входного файла: h.in
 Имя выходного файла: h.out
 Максимальное время работы на одном тесте: 1 секунда
 Максимальный объем используемой памяти: 64 мегабайта

Вы решили заказать пиццу с доставкой на дом. Известно, что для клиентов, сделавших заказ на сумму более C рублей, доставка является бесплатной, при заказе на C рублей и меньше доставка стоит B рублей.

Вы уже выбрали товар, стоимостью A рублей. В наличии имеются еще N товаров стоимостью d_1, \dots, d_N рублей, каждый в единственном экземпляре. Их также можно включить в заказ.

Как потратить меньше всего денег и получить на дом уже выбранный товар в A рублей?

Формат входных данных

Сначала вводятся числа A, B, C, N , а затем N чисел d_1, \dots, d_N .

Все числа целые, $1 \leq A \leq 1000$, $1 \leq B \leq 1000$, $1 \leq C \leq 1000$, $0 \leq N \leq 1000$, $1 \leq d_i \leq 1\,000\,000$.

Формат выходных данных

Выведите сначала суммарное количество денег, которое придется потратить. Если при этом вы планируете сделать дополнительный заказ с расчетом на бесплатную доставку, то далее выведите количество этих товаров и их номера в возрастающем порядке. Если же Вы будете оплачивать доставку сами, то далее выведите одно число -1 (минус один).

Примеры

h.in	h.out
10 17 25 5 2 7 5 3 7	26 3 1 2 5
100 1 50 5 5 2 4 3 1	100 0
10 14 25 5 2 7 5 3 7	24 -1

Задача I. Электронная очередь

Имя входного файла:	i.in
Имя выходного файла:	i.out
Максимальное время работы на одном тесте:	1 секунда
Максимальный объем используемой памяти:	64 мегабайта

В кассовом зале вокзала внедрили систему «электронная очередь». Теперь пассажир, который хочет купить билеты, при входе в зал получает талончик, после чего ожидает, когда подойдет очередь, и его пригласят к освободившейся кассе.

Система «электронная очередь» работает следующим образом. В зале имеются N касс, которые работают все время, за исключением времени перерывов (время перерывов в каждой кассе свое). Пассажиры приглашаются к кассам строго в порядке получения ими талончиков. Каждый пассажир проводит у кассы 5 минут (которые уходят на выбор поезда и оплату заказа), плюс оформление каждого билета дополнительно занимает еще одну минуту (таким образом, если пассажир, например, хочет оформить 3 билета, то он проведет у кассы 8 минут). Если пассажир был приглашен к кассе в момент времени A , а оформление его займет K минут, то и он, и эта касса освободятся к моменту времени $A+K+1$.

В освободившуюся кассу направляется очередной покупатель, если только в этой кассе не должно быть перерыва в ближайшие 5 минут. Но если касса начала оформление заказа какого-то пассажира, то она завершает оформление этого заказа, даже если в ней в процессе оформления должен начаться перерыв. При этом перерыв не сдвигается (то есть начинается на некоторое время позднее, но заканчивается ровно во столько, во сколько и должен).

Считается, что если перерыв в кассе с момента A до момента B , то касса, освобождаясь в момент $(A-5)$ начинает обслуживание очередного покупателя, а освобождаясь в моменты $(A-4)$, $(A-3)$, и так далее, — не начинает, а первого покупателя после перерыва касса начнет обслуживать в момент $(B+1)$.

Если одновременно освобождается сразу несколько касс, то первый по очереди попадает в кассу с меньшим номером, второй — в следующую и т.д. Если в момент, когда касса освобождается, в очереди нет пассажиров, касса, естественно, никого не обслуживает. В этом случае, как только новый пассажир приходит и встает в очередь, он тут же попадает на обслуживание в свободную кассу с минимальным номером.

Напишите программу, которая по информации о времени прихода каждого пассажира и о количестве билетов, которые он собирается купить, выведет время, когда этот пассажир покинет кассовый зал.

Ограничения системы «электронная очередь»

Ни один пассажир не покупает больше 10 билетов, никакие два пассажира не приходят в кассовый зал в один и тот же момент времени, между концом одного перерыва и началом следующего перерыва у одной кассы проходит не менее 60 минут, длительность каждого перерыва не менее 15 минут.

Формат входных данных

Сначала вводятся количество касс N и количество пассажиров M ($1 \leq N \leq 30$, $1 \leq M \leq 10000$). Далее идет описание перерывов каждой кассы: оно начинается с числа C_k ($1 \leq k \leq N$) — количество перерывов у данной кассы ($0 \leq C_k \leq 5$), а далее идут C_k пар чисел, задающих время начала перерыва и его длительность. Перерывы каждой кассы перечислены в порядке наступления.

Далее заданы M пар чисел, задающих время прихода очередного пассажира и количество билетов, которые он собирается купить. Пассажиры перечислены в порядке их прихода.

Все времена и длительности — натуральные числа, не превышающие 100000.

Формат выходных данных

Для каждого пассажира выведите два числа — номер кассы, которая будет его обслуживать, и время, в которое он покинет кассовый зал, купив нужные ему билеты.

Примеры

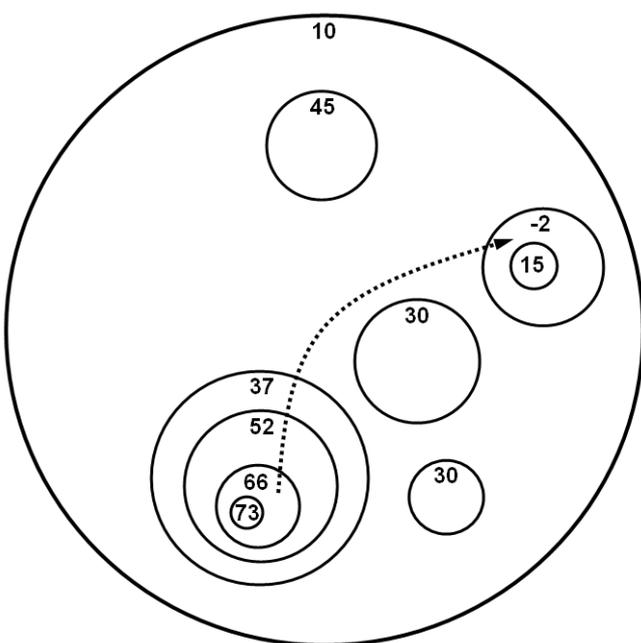
i.in	i.out	Комментарий
1 2 1 10 15 1 1 2 2	1 7 1 32	Имеется единственная касса с перерывом с 10-ю по 24-ю минуты. Первый пассажир будет сразу обслужен, его оформление закончится в 7 минут. За это время придет второй пассажир, однако до перерыва единственной кассы останется 3 минуты, поэтому она не будет никого принимать до его окончания. Второй пассажир будет принят только на 25 минуте и освободится в 32.
2 7 2 10 15 305 15 1 5 15 1 1 2 2 100 3 101 1 300 1 301 1 302 1	1 7 2 27 1 108 2 107 1 306 2 307 2 313	Первый пассажир сразу пойдет в первую кассу, она оформит его к 7-й минуте и не будет никого принимать до 25-й минуты в связи с перерывом. По той же причине вторая касса не сможет никого принять до окончания ее перерыва. На 20-й минуте к ней поступит второй пассажир. К 100-й минуте обе кассы будут свободны, поэтому пришедших в 100-ю и 101-ю минуты сразу пригласят к кассам (третьего – к минимальной по номеру свободной). На 300-й минуте пятый пассажир начнет оформляться в первой кассе, после чего она сразу закроется на перерыв до 320-й минуты. Шестому и седьмому пассажиру не остается ничего, кроме как обращаться во вторую кассу.

Задача J. Санная трасса

Имя входного файла: j.in
 Имя выходного файла: j.out
 Максимальное время работы на одном тесте: 1 секунда
 Максимальный объем используемой памяти: 64 мегабайта

Как известно, к северу от Москвы находится много горнолыжных трасс, расположенных на холмах Клинско-Дмитровской гряды. Один из курортов в связи с финансовым кризисом решил расширить спектр услуг, предлагая трассы для катания не только на лыжах и сноубордах, но и санные трассы.

У хозяев курорта имеется топографическая карта территории, высоты на



которой отображены с помощью контуров, каждый из которых представляет собой окружность. У каждой окружности указана высота поверхности, прилегающей к внутреннему контуру этой окружности. Вся территория, которая не находится внутри какой-либо окружности, имеет высоту 0. Поскольку это единственная информация о местности, то можно условно считать, что участки между окружностями плоские. Никакие две окружности не пересекаются и не касаются.

Используя эту карту, необходимо проложить санную трассу, которая будет удовлетворять двум условиям: разница высот между начальной и конечной точками должна

быть максимальна, и количество пересекаемых контуров не должно превышать некоторого заданного значения K (это связано с тем, что то место, которым сидят на санках, имеет ограниченную прочность). При этом трасса может иметь участки подъема, но не должна включать в себя ни одной точки, которая была бы выше начальной (туда санки просто не заедут).

На приведенном рисунке пунктирной линией показана наилучшая трасса для $K = 4$. Разница высот в ней составляет 68.

Формат входных данных

Сначала вводятся два натуральных числа C ($1 \leq C \leq 2\,000$) — количество окружностей и K ($1 \leq K \leq 2\,000$) — максимальное количество окружностей, которое может пересечь трасса.

Далее идут описания окружностей, каждое из которых состоит из четырех целых чисел: X, Y ($-2000 \leq X \leq 2000, -2000 \leq Y \leq 2000$) — координаты центра окружности, R ($1 \leq R \leq 2000$) — радиус окружности и A ($-1000 \leq A \leq 1000$) — высота местности, касающейся внутреннего края окружности.

Формат выходных данных

Выведите одно число — максимальный перепад высот на трассе.

Пример

j.in	j.out
10 4	68
38 61 2 73	
69 34 3 15	
61 59 4 30	
40 60 5 66	
58 44 6 30	
71 34 6 -2	
47 21 6 45	
41 58 8 52	
41 57 11 37	
48 40 33 10	