Оглавление

[Общие правила: 2](#_Toc21274788)

[1. Волосатый бизнес 3](#_Toc21274789)

[2. Зайчик 4](#_Toc21274790)

[3. Код 5](#_Toc21274791)

[4. Дачники 6](#_Toc21274792)

[5. Магараджа 7](#_Toc21274793)

[6. Баланс скобок 8](#_Toc21274794)

[7. Поиск прямоугольников 9](#_Toc21274795)

[8. Скобочки - 2 10](#_Toc21274796)

[9. ABC Path 11](#_Toc21274797)

[10. Калькулятор 13](#_Toc21274798)

[11. Задачка про лайки 14](#_Toc21274799)

[12. Усложненная (но более жизненная) задача про лайки 15](#_Toc21274800)

[13. Задачка про кинотеатр 16](#_Toc21274801)

## Общие правила:

* Каждая задача в отдельном файле
* Каждая задача представляет собой функцию с названием taskN, где N – номер в списке задач.
* Файлы Input.txt, Output.txt находятся рядом с файлом программы.
* 9 задача должна иметь сигнатуру: task9(grid) и возвращать целое число. Grid – string[] (массив/лист строк)
* 10 задача должна иметь сигнатуру task10(line) и в файл записывать строку с ответом или «Ошибка!» (без кавычек).

## Волосатый бизнес

*(Время: 1 сек. Память: 16 Мб Сложность: 32%)*

Одного неформала выгнали с работы, и теперь ему надо как-то зарабатывать себе на жизнь. Поразмыслив, он решил, что сможет иметь очень неплохие деньги на продаже собственных волос. Известно, что пункты приема покупают волосы произвольной длины стоимостью С у.е. за каждый сантиметр. Так как волосяной рынок является очень динамичным, то цена одного сантиметра волос меняется каждый день как и курс валют. Неформал является очень хорошим бизнес-аналитиком. Он смог вычислить, какой будет цена одного сантиметра волос в каждый из ближайших N дней (для удобства пронумеруем дни в хронологическом порядке от 0 до N-1). Теперь он хочет определить, в какие из этих дней ему следует продавать волосы, чтобы по истечению всех N дней заработать максимальное количество денег. Заметим, что волосы у неформала растут только ночью и вырастают на 1 сантиметр за ночь. Следует также учесть, что до 0-го дня неформал с горя подстригся наголо и к 0-му дню длина его волос составляла 1 сантиметр.

Входные данные

В первой строке входного файла INPUT.TXT записано целое число N (0 < N ≤ 100). Во второй строке через пробел заданы N натуральных чисел, не превосходящих 100, соответствующие стоимости C[i] 1 сантиметра волос за каждый i-й день.

Выходные данные

В единственную строку выходного файла OUTPUT.TXT нужно вывести максимальную денежную сумму, которую может заработать неформал за N дней.

Примеры

|  |  |  |
| --- | --- | --- |
| **№** | **INPUT.TXT** | **OUTPUT.TXT** |
| 1 | 5 73 31 96 24 46 | 380 |
| 2 | 10 1 2 3 4 5 6 7 8 9 10 | 100 |
| 3 | 10 10 9 8 7 6 5 4 3 2 1 | 55 |

## Зайчик

*(Время: 1 сек. Память: 16 Мб Сложность: 55%)*

|  |  |
| --- | --- |
| В нашем зоопарке появился заяц. Его поместили в клетку, и чтобы ему не было скучно, директор зоопарка распорядился поставить в его клетке лесенку. Теперь наш зайчик может прыгать по лесенке вверх, перепрыгивая через ступеньки. Лестница имеет определенное количество ступенек N. Заяц может одним прыжком преодолеть не более К ступенек. Для разнообразия зайчик пытается каждый раз найти новый путь к вершине лестницы. Директору любопытно, сколько различных способов есть у зайца добраться до вершины лестницы при заданных значениях K и N. Помогите директору написать программу, которая поможет вычислить это количество. Например, если K=3 и N=4, то существуют следующие маршруты: 1+1+1+1, 1+1+2, 1+2+1, 2+1+1, 2+2, 1+3, 3+1. Т.е. при данных значениях у зайца всего 7 различных маршрутов добраться до вершины лестницы. | https://acmp.ru/asp/article/image.asp?id=146 |
|  |  |

Входные данные

В единственной строке входного файла INPUT.TXT записаны два натуральных числа K и N (1 ≤ K ≤ N ≤ 300). К - максимальное количество ступенек, которое может преодолеть заяц одним прыжком, N – общее число ступенек лестницы.

Выходные данные

В единственную строку выходного файла OUTPUT.TXT нужно вывести количество возможных вариантов различных маршрутов зайца на верхнюю ступеньку лестницы без ведущих нулей.

Примеры

|  |  |  |
| --- | --- | --- |
| **№** | **INPUT.TXT** | **OUTPUT.TXT** |
| 1 | 1 3 | 1 |
| 2 | 2 7 | 21 |
| 3 | 3 10 | 274 |

## Код

*(Время: 1 сек. Память: 16 Мб Сложность: 36%)*

В наши дни в космосе находятся сотни спутников, и все они обмениваются данными. При этом система распознавания сигналов работает по схеме «Свой-Чужой». Один из спутников отправляет запрос другому спутнику в формате двух целых чисел, а второй спутник отвечает первому так же двумя целыми числами. Первые два числа первого спутника представляют собой сумму цифр и количество цифр тех двух чисел, которыми должен ответить второй спутник. При этом в качестве ответа должны получиться числа, представляющие наибольшее и наименьшее возможные значения, которые могут быть сформированы по описанному выше методу.

Вам предстоит написать программу, формирующую ответ для второго спутника по известным числам, полученным от первого спутника.

Входные данные

Во входном файле INPUT.TXT записаны 2 натуральных числа S и K, представляющих сумму и количество цифр соответственно (K ≤ 100). При этом гарантируется, что возможно составить хотя бы одно K-значное число, сумма цифр которого равна S.

Выходные данные

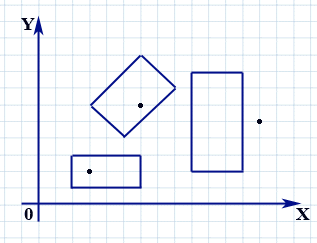
В выходной файл OUTPUT.TXT выведите два числа – ответ второго спутника. При этом следует помнить, что все числа не имеют лидирующих нулей.

Примеры

|  |  |  |
| --- | --- | --- |
| **№** | **INPUT.TXT** | **OUTPUT.TXT** |
| 1 | 1 3 | 100 100 |
| 2 | 2 3 | 200 101 |
| 3 | 3 4 | 3000 1002 |

## Дачники

*(Время: 1 сек. Память: 16 Мб Сложность: 45%)*

Всем известно, что дачники – народ странный, почти такой же, как и программисты. Строят они свои дачи непонятно где, да и выращивают там непонятно что и непонятно зачем. А уж как они туда добираются, это другая история: кто на автобусе, кто на электричке, кто на автомобиле, ну а кто-то вовсе пешком ходит от дома и до самого участка. Так что не стоит удивляться, если вдруг Вы узнаете, что некое садоводческое товарищество располагается на острове, а дачники добираются до него самолетом. Да еще и на этом острове может не быть посадочной полосы, так что высадиться на остров можно, только прыгая с парашютом (мы уж не рассматриваем то, как они возвращаются с дач домой). Рассмотрим этот уникальный случай. Пилот всегда старается осуществить высадку парашютистов таким образом, чтобы дачники приземлялись как можно ближе к своим прямоугольным участкам. Пилоту интересно знать: сколько дачников приземлится на свои участки? Помогите ему решить эту задачу!

Входные данные

В первой строке входного файла INPUT.TXT записано натуральное число N (1 ≤ N ≤ 1000) – количество дачников, далее идут N строк, в каждой из которых описаны координаты каждого дачника и его участка:  
X Y X1 Y1 X2 Y2 X3 Y3 X4 Y4  
где  
(X,Y) – координаты приземления парашютиста  
(X1, Y1, X2, Y2, X3, Y3, X4,Y4) – координаты прямоугольного участка на плоскости, указанные последовательно.  
Все координаты – целые числа, не превышающие 50000 по абсолютной величине

Выходные данные

В выходной файл OUTPUT.TXT нужно вывести количество дачников, приземлившихся на свой участок. Попадание на границу участка считается попаданием на участок.

Пример

|  |  |  |
| --- | --- | --- |
| **№** | **INPUT.TXT** | **OUTPUT.TXT** |
| 1 | 3 6 6 3 6 6 9 8 7 5 4 13 5 9 2 9 8 12 8 12 2 3 2 2 1 2 3 6 3 6 1 | 2 |

## Магараджа

*(Время: 1 сек. Память: 16 Мб Сложность: 60%)*

Магараджа — это шахматная фигура, сочетающая возможности ферзя и коня. Таким образом, магараджа может ходить и бить на любое количество клеток по диагонали, горизонтали и вертикали (т.е. как ферзь), а также либо на две клетки по горизонтали и на одну по вертикали, либо на одну по горизонтали и на две по вертикали (как конь).

Ваша задача — найти число способов расставить на доске N на N ровно K магараджей так, чтобы они не били друг друга.

Входные данные

Входной файл INPUT.TXT содержит два целых числа: N и K (1 ≤ K ≤ N ≤ 10).

Выходные данные

В выходной файл OUTPUT.TXT выведите ответ на задачу.

Примеры

|  |  |  |
| --- | --- | --- |
| **№** | **INPUT.TXT** | **OUTPUT.TXT** |
| 1 | 3 1 | 9 |
| 2 | 4 2 | 20 |
| 3 | 5 3 | 48 |

# Баланс скобок

*(Время: 1 сек. Память: 16 Мб Сложность: 42%)*

Дана последовательность, состоящая из открывающихся и закрывающихся круглых, квадратных и фигурных скобок.

Требуется написать программу, которая определит возможность добавления в эту последовательность цифр и знаков арифметических действий таким образом, чтобы получилось правильное скобочное выражение.

Входные данные

Входной файл INPUT.TXT содержит не менее 1 и не более 10 строк. В каждой строке записана одна последовательность скобок. Длина последовательности от 1 до 255 символов.

Выходные данные

В выходной файл OUTPUT.TXT выведите слитно символы 0 или 1. Их общее количество равно количеству введенных строк. Для каждой строки выводится 0, если из нее может получиться правильное скобочное выражение, и 1 иначе.

Пример

|  |  |  |
| --- | --- | --- |
| **№** | **INPUT.TXT** | **OUTPUT.TXT** |
| 1 | ([{}]) ([{ | 01 |

## Поиск прямоугольников

*(Время: 0,5 сек. Память: 16 Мб Сложность: 34%)*

На поле N×M клеток (N строк и M столбцов) положили K прямоугольников один поверх другого в случайном порядке. Длины сторон прямоугольников выражаются целым числом клеток. Прямоугольники не выходят за границы поля. Границы прямоугольников совпадают с границами клеток поля.

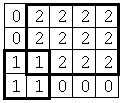
Получившуюся ситуацию записали в таблицу чисел (каждой клетке поля соответствует клетка таблицы). Если клетка поля не закрыта прямоугольником, то в соответствующую клетку таблицы записали число 0. Если же клетка закрыта одним или несколькими прямоугольниками, то в соответствующую клетку таблицы записали число, соответствующее номеру самого верхнего прямоугольника, закрывающего эту клетку.

Требуется написать программу, которая определит положение и размеры прямоугольников. Гарантируется, что во входных данных содержится информация, которой достаточно для однозначного определения размеров прямоугольников.

Входные данные

Входной файл INPUT.TXT содержит в первой строке целые числа N, M, K (1 ≤ N ≤ 200, 1 ≤ M ≤ 200, 1 ≤ K ≤ 255). Далее следует N строк по M чисел в каждой — содержимое таблицы. Все числа в таблице целые, находятся в диапазоне от 0 до K включительно.

Выходные данные

****Выходной файл OUTPUT.TXT должен содержать K строк. Каждая строка должна описывать соответствующий ее номеру прямоугольник четырьмя числами X1 Y1 X2 Y2 (X1 и Y1 должны описывать координаты левого нижнего угла прямоугольника, а X2 и Y2 — координаты правого верхнего угла). Числа должны разделяться пробелом.

Начало координат расположено в левом нижнем углу таблицы. Таким образом, координаты левого нижнего угла поля — (0,0), правого верхнего — (M,N).

Пример

|  |  |  |
| --- | --- | --- |
| **№** | **INPUT.TXT** | **OUTPUT.TXT** |
| 1 | 4 5 2 0 2 2 2 2 0 2 2 2 2 1 1 2 2 2 1 1 0 0 0 | 0 0 2 2 1 1 5 4 |

## Скобочки - 2

*(Время: 1 сек. Память: 16 Мб Сложность: 37%)*

Напомним, что называется правильной скобочной последовательностью:

* пустая строка является правильной скобочной последовательностью;
* если строка а — правильная скобочная последовательность, то строки (а), [а] — тоже правильные скобочные последовательности;
* если строки а и b — правильные скобочные последовательности, то строка ab — тоже правильная скобочная последовательность.

Задана строка S, состоящая из квадратных и круглых скобок. Разрешается заменять квадратную открывающую скобку ([) на круглую открывающую (() и наоборот, а также квадратную закрывающую скобку (]) на круглую закрывающую ()) и наоборот.

За одно действие разрешается изменить ровно один символ строки. Необходимо за минимальное число действий преобразовать S в правильную скобочную последовательность.

Входные данные

Входной файл INPUT.TXT содержит строку S. Ее длина не превосходит 100 000 символов.

Выходные данные

В выходной файл OUTPUT.TXT выведите искомое минимальное число действий или -1, если преобразовать S в правильную скобочную последовательность невозможно.

Примеры

|  |  |  |
| --- | --- | --- |
| **№** | **INPUT.TXT** | **OUTPUT.TXT** |
| 1 | (())[] | 0 |
| 2 | [(]) | 2 |
| 3 | ((]]] | -1 |

## ABC Path

--Problem Statement--

You will be given a 2-dimensional grid of letters. Write a method to find the length of the longest path of consecutive letters, starting at 'A'. Paths can step from one letter in the grid to any adjacent letter (horizontally, vertically, or diagonally).

For example, in the following grid, there are several paths from 'A' to 'D', but none from 'A' to 'E':

{ "ABE",

"CFG",

"BDH",

"ABC" }

One such path is:

A B .

C . .

. D .

. . .

(spaces are for clarity only)

so, for this grid, your method should return 4.

--Definition--

Class: ABCPath

Method: length

Parameters: string[]

Returns: int

Method signature: int length(string[] grid)

(be sure your method is public)

--Limits--

Time limit (s): 840.000

Memory limit (MB): 64

--Notes--

- The longest path may start at any 'A' character in the input.

Constraints

- grid will contain between 1 and 50 elements, inclusive.

- Each element of grid will be between 1 and 50 characters long, inclusive.

- Each element of grid will have the same length.

- grid will contain only uppercase letters ('A'-'Z').

--Examples--

0)

{ "ABE", "CFG", "BDH", "ABC" }

Returns: 4

This is the example from the problem statement.

1)

{ "A" }

Returns: 1

2)

{ "BCDEFGHIJKLMNOPQRSTUVWXYZ" }

Returns: 0

Paths must start with an 'A'.

## Калькулятор

Написать калькулятор, которые принимает на вход строку, введенную пользователем.

Строка может содержать следующие символы: 0-9, (), + - \* /

Если строка представляет собой верное арифметическое выражение, то вывести результат, иначе сообщить об ошибке. Все числа в строке – целые.

Примеры:

1+2-(3-5)/2 = 4

-10/2+5 = 0

2+2\*2 = 6

2\*(2+2) = 8

-2++3( = Ошибка!

|  |  |
| --- | --- |
| 11 |  |
| 12 |  |

<https://www.hackerrank.com/domains/sql?filters%5Bstatus%5D%5B%5D=unsolved&badge_type=sql>