Указания по решению заданий II (муниципального) этапа всероссийской олимпиады школьников по информатике в 2016/2017 учебном году

Комплект заданий для 7-8 классов

Ограничения на время работы решений на одном тесте, а также ограничения на объём используемой памяти в задачах на программирование указаны на соответствующих вкладках в тестирующей системе.

Задание 1. Перестановка

Вам даны пять чисел:
1890
83849
126631
146531
32749972

Для каждого найдите минимальное целое число, которое больше данного и состоит из такого же набора цифр. Например, для числа 1932 это 2139, а для числа 1212 — это 1221.

В ответ запишите пять целых чисел в пяти отдельных строках. Порядок записи менять нельзя. Если вы не можете найти ответ для какого-то числа, то на соответствующей строке поставьте знак «-» (минус без кавычек). Все числа равнозначны и оцениваются в 20 баллов.

Задание 2. Лягушка

Вдоль прямой расположены 15 кочек. На первой сидит лягушка, которая хочет попасть на пятнадцатую. Одним прыжком лягушка может прыгнуть вправо на следующую кочку или через одну кочку. Пятая и девятая кочки кажутся лягушке подозрительными, поэтому на них она прыгать не будет (только через них).

Л●	2●	3●	4●	—	6●	7●	8●	—	10●	11●	12●	13●	14●	15●

Ответьте на пять вопросов:

1. Какое максимальное число прыжков надо сделать, чтобы оказаться на 15-й кочке?
2. Какое минимальное число прыжков надо сделать, чтобы оказаться на 15-й кочке?
3. Сколько существует различных способов добраться с первой кочки до 15-й, совершив минимальное количество прыжков?
4. Сколько всего существует различных способов добраться с первой кочки до 15-й?
5. Рассмотрим все различные способы добраться с первой кочки до пятнадцатой. Сколько из них проходят через тринадцатую кочку?

В ответ запишите пять целых чисел в пяти отдельных строках. Порядок записи менять нельзя. Если вы не можете найти ответ на какой-то вопрос, то на соответствующей строке поставьте знак «-» (минус без кавычек). Все вопросы равнозначны и оцениваются в 20 баллов.

Задание 3. Абацабадабацаба

Возьмём строку **a**, и преобразуем по такому правилу: запишем её после самой себя, но при этом между двумя половинами поставим первый символ английского алфавита из тех, которые ещё не встречались в строке (то есть **b**). Получается a**b**a. Повторим преобразование много раз: a**b**a → aba**c**aba → abacaba**d**abacaba → abacabadabacaba**e**abacabadabacaba и т.д.

Вам даны пять чисел:
20
48
128
491520
12535808

Для каждого из них найдите какая буква идёт в строке на этой позиции. Напомним, что в английском алфавите 26 букв abcdefghijklmnopqrstuvwxyz.

В ответ запишите пять строчных (маленьких) английских символов в пяти отдельных строках. Порядок записи менять нельзя. Если вы не можете найти ответ для какого-то числа, то на соответствующей строке поставьте знак «-» (минус без кавычек). Все числа равнозначны и оцениваются в 20 баллов.

Задание 4. Зарядка

Ваня ведёт дневник, где в том числе отмечает, делал он зарядку в этот день или нет. Напишите программу, которая по собранной статистике сообщит, сколько последних дней подряд Ваня делал зарядку, а так же наибольшее число дней подряд когда он делал зарядку.

Формат входных данных

Дана строка, в которой каждый символ соответствует одному дню. Знаком + отмечены дни, когда Ваня делал зарядку, точкой когда нет. Длина строки меньше 10^5 символов

Формат выходных данных

Выведите через пробел два числа.

Примеры

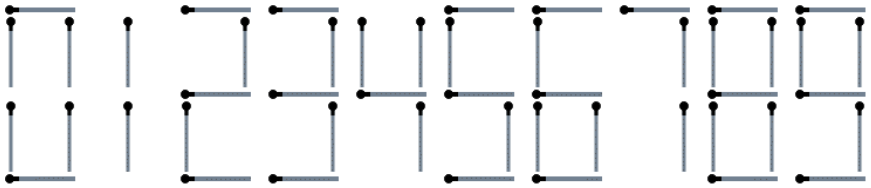
входные данные	результат
`+++`	`3 3`
`++.++++...+`	`1 4`

Система оценки

В задаче 50 тестов. За прохождение каждого теста даётся 2 балла. В тридцати тестах длина строки меньше ста. В сорока тестах длина строки меньше тысячи.

Задание 5. Числа из спичек

Из спичек можно выкладывать цифры следующим образом:



Напишите программу, которая для заданного числа определит, сколько понадобится спичек, чтобы выложить все цифры этого числа.

Формат входных данных

В первой строке записано неотрицательное целое число X. Гарантируется, что количество цифр в числе меньше ста.

Формат результата

Выведите одно целое число — количество спичек.

Примеры

входные данные	результат
`1`	`2`
`239`	`16`

Система оценки

Тесты к задаче состоят из трёх групп. В первой тесты оцениваются независимо по два балла за тест. Во второй и третьей тесты оцениваются независимо по четыре балла за тест.

Группа	Тесты	Баллы	Комментарий
1	1-10	20	$X < 10$
2	11-20	40	$X < 10^5$
3	21-30	40	

Комплект заданий для 9-11 классов

Ограничения на время работы решений на одном тесте, а также ограничения на объём используемой памяти указаны на вкладках задач в тестирующей системе.

Задача 1. Зарядка

Ваня ведёт дневник, где в том числе отмечает, делал он зарядку в этот день или нет. Напишите программу, которая по собранной статистике сообщит, сколько последних дней подряд Ваня делал зарядку, а так же наибольшее число дней подряд когда он делал зарядку.

Формат входных данных

Дана строка, в которой каждый символ соответствует одному дню. Знаком + отмечены дни, когда Ваня делал зарядку, точкой когда нет. Длина строки меньше 10^5 символов

Формат выходных данных

Выведите через пробел два числа.

Примеры

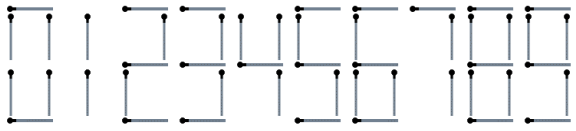
входные данные	результат
+++	3 3
++.++++...+	1 4

Система оценки

В задаче 50 тестов. За прохождение каждого теста даётся 2 балла. В тридцати тестах длина строки меньше ста. В сорока тестах длина строки меньше тысячи.

Задача 2. Числа из спичек

Из спичек можно выкладывать цифры следующим образом:



Напишите программу, которая для заданного числа определит, сколько понадобится спичек, чтобы выложить все цифры этого числа.Ф*ормат входных данных*

Формат входных данных

В первой строке записано неотрицательное целое число *X*. Гарантируется, что количество цифр в числе меньше ста.

Формат результата

Выведите одно целое число — количество спичек.

Примеры

входные данные	результат
1	2
239	16

Система оценки

Тесты к задаче состоят из трёх групп. В первой тесты оцениваются независимо по два балла за тест. Во второй и третьей тесты оцениваются независимо по четыре балла за тест.

Группа	Тесты	Баллы	Комментарий
1	1-10	20	$X < 10$
2	11-20	40	$X < 10^5$
3	21-30	40	

Задача 3. Abacabadabacaba

Возьмём строку **a**, и преобразуем по такому правилу: запишем её после самой себя, но при этом между двумя половинами поставим первый символ английского алфавита из тех, которые ещё не встречались в строке (то есть **b**). Получается a**b**a. Повторим преобразование много раз: a**b**a → aba**c**aba → abacaba**d**abacaba и т.д. Если в строке есть все строчные буквы английского алфавита, то ставится прописная буква A и далее B, C, D, E...

Напишите программу, которая находит букву на позиции *N*. Напомним, что в английском алфавите 26 букв abcdefghijklmnopqrstuvwxyz.

Формат входных данных

В первой строке записано целое число N ($1 \le N < 10^9$).

Формат результата

Выведите букву английского алфавита.

Примеры

входные данные	результат
12	c

Система оценки

Группа	Тесты	Баллы	Комментарий
1	1-10	10	$N < 50$, тесты оцениваются независимо
2	11-20	30	$N < 1000$, тесты оцениваются независимо
3	21-30	30	$N < 10^6$, тесты оцениваются независимо
4	31-40	30	тесты оцениваются независимо

Задача 4. Анаграммы

Строка *A* называется анаграммой строки *B* если она состоит в точности из тех же букв, что и строка *B*, но составленных возможно в другом порядке. Например, строка heart является анаграммой строки earth. Другие анаграммы строки earth это earth, aehrt, ahter, trhea, ...

Напишите программу, которая для двух строк *T* и *P* определит, сколько раз анаграммы строки *P* встречаются в строке *T*.

Формат входных данных

Первая строка это *T*, вторая это *P*. Строки состоят из строчных букв английского алфавита. Длины строк меньше 10^5 символов.

Формат результата

Выведите сколько раз анаграммы строки *P* встречаются в строке *T*.

Примеры

входные данные	результат
tototoot toot	4

Примечание

В примере ответ четыре, так как два раза встречается строка toto и по одному разу строки otot и toot. Все они являются анаграммами строки toot.

Система оценки

Группа	Тесты	Баллы	Комментарий
1	1-10	60	длины строк меньше ста; тесты оцениваются независимо
2	11-15	40	тесты оцениваются независимо

Указания по решению задач

Перестановка

Необходимо перебирать цифры числа с конца и найти первую, которая может быть увеличена за счёт цифр справа от неё. Поскольку требуется найти минимальное число больше данного, то найденную цифру следует заменить на минимальную из больших, находящихся справа. Оставшиеся цифры упорядочиваются по возрастанию.

Например, в числе 648531 первая с конца цифра, которую можно заменить это четыре. Её можно заменить на пять или восемь. Нас интересует минимальный ответ, поэтому пять. Получается начало 65.... и оставшиеся цифры располагаются в порядке увеличения. В итоге 651348 это искомое число.

Лягушка

Максимальное число прыжков получается, если стараться прыгать всегда на соседнюю кочку и только если это невозможно, то прыгать через одну. Аналогично минимальное число прыжков получается если прыгать всегда через одну и только если это невозможно, то прыгать на соседнюю.

Кочки пять и девять разбивают прямую на три участка: длины четыре (с первой по четвёртую), длины три (с шестой по восьмую) и длины шесть (с десятой по пятнадцатую). Эти участки можно рассматривать отдельно, так как лягушка обязана оказаться на кочках четыре и шесть (чтобы перепрыгнуть пятую) и на кочках восемь и десять (чтобы перепрыгнуть девятую).

Минимальный путь с первой до четвёртой занимает два прыжка. И таких вариантов два (1→3→4, 1→2→4). Минимальный путь с шестой до восьмой занимает один прыжок и такой вариант ровно один. Минимальный путь с десятой до пятнадцатой занимает три прыжка и таких вариантов три (10→11→13→15, 10→12→13→15, 10→12→14→15). Так как участки независимы, то всего вариантов шесть ($6 = 2 \times 1 \times 3$).

Чтобы найти количество различных способов допрыгать с первой до пятнадцатой, необходимо последовательно найти это количество для всех кочек, начиная со второй. На вторую кочку можно попасть с первой и такой способ ровно один. На третью кочку можно попасть с первой (один способ) и со второй (один способ) и всего таких способов два. На четвёртую можно попасть со второй (один способ) и с третьей (два способа) и всего таких способов три и т.д.

Чтобы найти количество способов, содержащих тринадцатую клетку, можно из общего количества, найденного на предыдущем шаге, вычесть количество способов в которых тринадцатой клетки нет. Это число получается, если запретить лягушке прыгать на тринадцатую клетку (сделать её как пятая и девятая) и найти количество способов в этом случае.

Абацабадабацаба

Для первых трёх чисел ответ можно найти если построить последовательность вручную в текстовом редакторе. Для четвёртого и пятого чисел можно написать вспомогательную программу.

Так же ответы находятся вручную, если заметить, что по построению буква a всегда стоит на нечётных позициях, буква b – на чётных позициях, но номера, которых не делятся на четыре, буква c – на позициях, номера которых делятся на четыре, но не делятся на восемь. Это обобщается. При каждом преобразовании, появляется позиция, на которую ставится новая буква, и номер этой позиции это степень двойки. По построению, в дальнейшем на всех позициях, номера которых делятся на эту степень двойки будет находится этот символ.

Тогда решение это найти максимальную степень двойки на которую делится число и ответ это соответствующая буква английского алфавита при нумерации с нуля. Например, на позиции 3824 находится буква e так, как 3824 делится на 16 и не делится на 32. Шестнадцать это два в четвёртой степени и четвёртая буква английского алфавита при нумерации с нуля это e.

Зарядка

Задача на работу со строками. Длину последней группы из символов + можно узнать, если найти самую правую точку в строке. Тогда всё что правее неё — это последняя подстрока из символов +. Максимальную подстроку из символов + можно найти за один проход, если завести переменную в которой считать сколько символов + подряд мы уже просмотрели. Символ + увеличивает эту переменную на один, точка сбрасывает в ноль. В процессе необходимо сравнивать максимальное найденное значение с текущим и обновлять при улучшении.

Числа из спичек

По картинке необходимо определить сколько спичек требуется для каждой цифры. После этого остаётся просмотреть все цифры числа и для каждой прибавить к ответу соответствующее количество. Так как входные данные могут быть большими, то рекомендуется читать входное число как строку.

Abacabadabacaba

Это задача «Абацабадабацаба», но с одним отличием, которое состоит в том, что алгоритм необходимо реализовать в виде программы, а не применить вручную для нескольких частных случаев.

```
aA = "abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ"
n = int(input())
k = 0
while n % 2 == 0:
  n = n / 2
  k += 1
print(aA[k])
```

Анаграммы

Шестьдесят баллов можно набрать, проверяя для каждой позиции в строке *T*, верно ли, что следующие символы это перестановка символов строки *P*. Проверить, что две строки *A* и *B* являются анаграммами на языке Питон можно сравнив sorted(A) и sorted(B). Пример решения на 60 баллов на Питоне:

```
t = input()
p = input()
a = sorted(p)
k = 0
for i in range(len(t) - len(p) + 1):
  b = sorted(t[i : i + len(p)])
  if a == b:
    k += 1
print(k)
```

Это решение правильное, но медленное для больших строк. Пример быстрого решения это двигать вдоль строки *T* «окно» такой же ширины, как и строка *P*. Будем хранить для каждой буквы сколько раз она встречается в «окне». При сдвиге у одной буквы счётчик уменьшиться на один, а у другой увеличиться на один. Если для каждой буквы заранее подсчитать сколько раз она встречается в строке *P*, то можно быстро обновлять информацию о том, сколько букв в «окне» встречаются или слишком мало раз или слишком много. Если это количество равно нулю, то текущий сдвиг содержит анаграмму. Детали реализации данного алгоритма можно увидеть в решении жюри.