

Комплект заданий для 7-8 классов

Все решения необходимо отправить на автоматическую проверку. Для этого используйте вкладку «Посылки» в проверяющей системе.

В задачах, где ответ это программа, время работы на одном тесте не должно превышать одной секунды. Ограничение на объём используемой памяти составляет 256 Мб.

Для решения задания с текстовым ответом разрешается использовать компьютер. Например, вам может понадобиться текстовый редактор, приложение «Калькулятор» или написание вспомогательной программы. Не спешите отправлять задание с текстовым ответом — его можно отправить на проверку не более **трех** раз (в отличие от заданий, где ответ это программа).

Задание 1. Игра

В компьютерной игре на клетчатую карту высотой h и шириной w клеток помещают капли воды так, что некоторые клетки становятся влажными. Далее влага начинает распространяться. Если в какой-то момент времени у сухой клетки есть хотя бы две влажных соседних клетки (соседние клетки имеют общую сторону), то она также становится влажной.

Для следующих размеров карты, определите, какое **минимальное** число капель нужно поместить на нее в начале так, чтобы в результате все клетки стали влажными.

1. $h=1$ $w=6$
2. $h=5$ $w=5$
3. $h=7$ $w=100$
4. $h=140$ $w=81$
5. $h=2018$ $w=239$

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

Задание 2. Какие-то числа

Найдите максимальное целое число длины n , среди всех чисел подходящих под определенное условие. Чтобы проверить, подходит число под условие или нет, необходимо рассмотреть его цифры, начиная со второй и до предпоследней. Если для каждой такой цифры сумма двух ее соседних цифр больше чем удвоенное значение самой цифры, то число подходит под условие.

Например, 51249 подходит так, как $5+2>1*2$, $1+4>2*2$, $2+9>4*2$. Примеры подходящих четырехзначных чисел это 1049, 7421, 2358, 2018 и т.д. Максимальное из них и соответственно ответ для $n=4$ это 9889.

Найдите максимальное число длины n , подходящее под описанное выше условие для следующих n :

1. $n=3$
2. $n=5$
3. $n=6$
4. $n=7$
5. $n=8$

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

Задание 3. Пять и шесть

Если двузначное число 76 возвести в квадрат, то получится 5776. Заметьте, что десятичная запись квадрата числа оканчивается самим числом. Аналогичным свойством обладает трехзначное число 625 ($625^2 = 390625$).

Для следующих n найдите положительное целое n -значное число, десятичная запись квадрата которого, оканчивается самим числом:

1. $n=1$
2. $n=4$
3. $n=13$
4. $n=25$
5. $n=130$

В ответ запишите пять чисел без ведущих нулей на пяти отдельных строках. Порядок записи менять нельзя. Если вы не можете найти ответ для какого-то вопроса, то на соответствующей строке поставьте знак «-» (минус без кавычек). Все вопросы равнозначны и оцениваются в 20 баллов. Если для какого-то n существует несколько правильных ответов, то можно выбрать любой.

Примечание. Число называется n -значным, если его десятичная запись состоит из n знаков. То есть 123456 это шестизначное число, а 9876 это четырехзначное. И тогда ответом на третий вопрос должно быть какое-то число, состоящее из тринадцати знаков такое, что десятичная запись квадрата этого числа заканчивается самим числом.

Задание 4. Обмен цифр

Даны два числа. За одну операцию можно обменять местами цифры на одной позиции в разных числах.

Например, пару чисел 98 и 765 за одну операцию можно преобразовать в пару 78 и 965, обменяв первые цифры или в пару 96 и 785, обменяв вторые цифры. Цифры на позиции три менять нельзя так, как в числе 98 всего две цифры.

Найдите минимальное количество операций с помощью которых можно сделать так, чтобы второе число стало больше первого.

Формат входных данных

В двух строках записаны целые положительные числа. Каждое число не превышает 10^9 .

Формат выходных данных

Если невозможно второе число сделать больше первого, то выведите -1. Иначе в первой строке выведите минимальное количество операций обмена. Во второй строке, разделяя пробелом, запишите позиции в которых необходимо произвести обмены.

Примеры

входные данные	результат
456 123	1 1
10 1	-1
1 10	0

Система оценки

Все тесты в задаче равнозначны и оцениваются в 2 балла.

Комплект заданий для 9-11 классов

Все решения необходимо отправить на автоматическую проверку. Для этого используйте вкладку «Посылки» в проверяющей системе.

В задачах, где ответ это программа, время работы на одном тесте не должно превышать одной секунды. Ограничение на объём используемой памяти составляет 256 Мб.

Для решения задания с текстовым ответом разрешается использовать компьютер. Например, вам может понадобиться текстовый редактор, приложение «Калькулятор» или написание вспомогательной программы. Не спешите отправлять задание с текстовым ответом — его можно отправить на проверку не более **трех** раз (в отличие от заданий, где ответ это программа).

Задание 1. Пять и шесть

Если двузначное число 76 возвести в квадрат, то получится 5776. Заметьте, что десятичная запись квадрата числа оканчивается самим числом. Аналогичным свойством обладает трехзначное число 625 ($625^2 = 390625$).

Для следующих n найдите положительное целое n -значное число, десятичная запись квадрата которого, оканчивается самим числом:

6. $n=1$
7. $n=4$
8. $n=13$
9. $n=25$
10. $n=130$

В ответ запишите пять чисел без ведущих нулей на пяти отдельных строках. Порядок записи менять нельзя. Если вы не можете найти ответ для какого-то вопроса, то на соответствующей строке поставьте знак «-» (минус без кавычек). Все вопросы равнозначны и оцениваются в 20 баллов. Если для какого-то n существует несколько правильных ответов, то можно выбрать любой.

Примечание. Число называется n -значным, если его десятичная запись состоит из n знаков. То есть 123456 это шестизначное число, а 9876 это четырехзначное. И тогда ответом на третий вопрос должно быть какое-то число, состоящее из тринадцати знаков такое, что десятичная запись квадрата этого числа заканчивается самим числом.

Задание 2. Обмен цифр

Даны два числа. За одну операцию можно обменять местами цифры на одной позиции в разных числах.

Например, пару чисел 98 и 765 за одну операцию можно преобразовать в пару 78 и 965, обменяв первые цифры или в пару 96 и 785, обменяв вторые цифры. Цифры на позиции три менять нельзя так, как в числе 98 всего две цифры.

Найдите минимальное количество операций с помощью которых можно сделать так, чтобы второе число стало больше первого.

Формат входных данных

В двух строках записаны целые положительные числа. Каждое число не превышает 10^9 .

Формат выходных данных

Если невозможно второе число сделать больше первого, то выведите -1. Иначе в первой строке выведите минимальное количество операций обмена. Во второй строке, разделяя пробелом, запишите позиции в которых необходимо произвести обмены.

Примеры

входные данные	результат
456 123	1 1
10 1	-1
1 10	0

Система оценки

Все тесты в задаче равнозначны и оцениваются в 2 балла.

Задание 3. Неразнообразные подстроки

Назовем строку, состоящую из строчных латинских букв и точек *разнообразной* если в ней встречаются хотя бы две различных буквы. Примеры, разнообразных строк: *abba* | *..qp.r.q* | *abcde* (здесь вертикальная черта выступает разделителем для удобства чтения). Примеры неразнообразных строк: *aaa* | *.....* | *.b...b*.

Найдите количество неразнообразных подстрок в данной строке *s*.

Формат входных данных

В первой строке записана строка *s*, состоящая из строчных латинских букв и точек. Гарантируется, что длина *s* не превышает 400000 символов.

Формат результата

Выведите количество искомых подстрок.

Примеры

входные данные	результат
b..d	9
a.bb.a	14

Примечание

Во втором примере подходят 14 подстрок: 6 подстрок длины один, 5 подстрок длины два (*a.* | *.b* | *bb* | *b.* | *.a*), 2 подстроки длины три (*.bb* | *bb.*) и 1 подстрока длины четыре (*.bb.*).

Система оценки

Тесты к задаче состоят из трёх групп. Тесты в первой группе оцениваются независимо по два балла за тест. Баллы за вторую и третью группу начисляются только при прохождении всей группы.

Группа	Баллы	Комментарий
1	40	длина $s \leq 300$
2	40	длина $s \leq 2500$
3	20	

Задание 4. Среднее арифметическое

На доске выписана последовательность из n целых чисел. Также известно число k . Вопрос: можно ли стереть ноль или больше чисел в начале последовательности и стереть ноль или больше чисел в конце последовательности так, чтобы среднее арифметическое оставшихся чисел было в точности равно k ?

Если ответ утвердительный, то необходимо найти способ в котором количество оставшихся чисел максимально.

Формат входных данных

В первой строке записаны целые числа n и k ($1 \leq n \leq 10^5$; $1 \leq k \leq 10^9$). Во второй строке через пробел указаны n положительных целых чисел, каждое из которых не превосходит 10^9 .

Формат результата

Если ответ на вопрос утвердительный, то выведите через пробел два числа — количество оставшихся чисел и номер первого оставшегося числа. В противном случае выведите ноль. Если решений с максимальным числом оставшихся элементов несколько, то выведите любое.

Примеры

входные данные	результат
4 3 3 1 2 4	2 3
4 4 3 1 2 4	1 4
4 5 3 1 2 4	0

Система оценки

Тесты к задаче состоят из трёх групп. Тесты в каждой группе оцениваются независимо по два балла за тест.

Группа	Баллы	Комментарий
1	30	$n \leq 100$
2	30	$n \leq 1000$
3	40	

Указания по решению задач

Игра

Для получения ответа необходимо разделить на два сумму h и w и округлить результат вверх до ближайшего целого.

Сначала покажем, что такого количество достаточно. Без уменьшения общности будем считать, что $h \leq w$. Тогда в квадрате $h \times h$ слева можно поместить капли на главной диагонали. Останется прямоугольник шириной $w-h$ и высотой h клеток. В нижней строке этого прямоугольника можно разместить капли воды в каждом втором столбце, начиная с правого.

Пример для 4×9 :	Пример для 4×8 :
+-----	+-----
-+-----	-+-----
--+-----	--+-----
---+--+--+	---+--+--+

Такое распределение приведет к тому, что все клетки карты станут влажными. Количество капель это $h+(w-h)/2=(h+w)/2$ и плюс один если $w-h$ нечетно.

То, что меньше капель разместить нельзя следует из следующего наблюдения. При появлении новых влажных капель периметр влажной области не изменяется. В конце этот периметр должен составлять $(h+w)*2$. Периметр одной капли равен четырем, поэтому начальное количество капель не может быть меньше $(h+w)*2/4=(h+w)/2$.

Правильные ответы: 4, 5, 54, 111, 1129

Какие-то числа

Можно перебирать варианты вручную. Можно написать программу, которая для каждой требуемой длины перебирает числа от максимального вниз и проверяет, подходит или нет. Пример проверки одного числа:

```
def good(x):
    d = [int(_) for _ in str(x)]
    for i in range(1, len(d)-1):
        if d[i] * 2 >= d[i-1] + d[i+1]:
            return False
    return True
```

Правильные ответы: 989, 97679, 976679, 9643469, 96433469

Пять и шесть

Если есть число длины L которое в квадрате заканчивается на само себя, то последние $L-1$, $L-2$, ...1 цифра также образуют число, которое в квадрате заканчивается на само себя. Например: $90625^2=8212890625$, $0625^2=390625$, ..., $5^2=25$).

Это так потому, что на $L-1$ последнюю цифру квадрата влияют только $L-1$ последние цифры самого числа (аналогично для $L-2$, $L-3$, ..., 1).

Этим фактом можно воспользоваться и зная однозначное число (например $6^2=36$), попробовать найти двузначное «в лоб», проверив все варианты 06, 16, 26, 36, ..., 96. Окажется, что $76^2=5776$. Теперь можно перейти к трехзначным числам и перебрать 076, 176, ..., 976: $376^2=141376$. С помощью программы это делается быстрее, чем вручную и продолжая вычисления можно найти число длины 130.

Всего существует два варианта 130-и значного числа:

18013756190554629968147642639039530073191081698029385098900621665095808
63811000557423423230896109004106619977392256259918212890625,

81986243809445370031852357360960469926808918301970614901099378334904191
36188999442576576769103890995893380022607743740081787109376.

Обмен цифр

Если первое число длиннее второго или они равны, то невозможно. Если второе число длиннее первого или больше, то ничего делать не надо. Иначе числа одинаковой длины и первое больше второго. Необходимо найти самую левую цифру, где числа отличаются. В этой позиции цифра первого числа больше и в ней нужно произвести обмен.

```
a = input()
b = input()

if len(a) > len(b) or a == b:
    print(-1)
elif len(b) > len(a) or b > a:
    print(0)
else:
    i = 0
    while a[i] == b[i]:
        i = i + 1
    print(1)
    print(i + 1)
```

Неразнообразные подстроки

На сорок баллов можно перебрать все строки и для каждой проверить является ли она неразнообразной.

```
def check(s):
    ...

s = input()
answer = 0
for i in range(len(s)):
    for j in range(i+1, len(s)+1):
        substring = s[i:j]
        if check(substring):
            answer += 1
print(answer)
```

Верно что, что если $s[i:j]$ – разнообразная, то и $s[i:j+1]$, $s[i:j+2]$, ..., $s[i:]$ – разнообразные. Поэтому если $check(substring)$ вернет True, то она и дальше будет возвращать True для всех больших j . Аналогично, если $s[i:j]$ – неразнообразная, то и все меньшие подстроки $s[i:j-1]$, $s[i:j-2]$, ..., $s[i:i+1]$ – неразнообразные. Это позволяет написать решение сложностью $O(n^2)$, которое набирает 80 баллов. Для каждого i необходимо найти такое j , что $s[i:j]$ – наибольшая неразнообразная строка, начинающаяся в i . И тогда из предыдущего наблюдения следует, что всего $j-i$ неразнообразных строки начинается в позиции i .


```
s = input()
answer = 0
for i in range(len(s)):
    j = i
    expected = None
    while j < len(s):
        if s[j] != "." and expected != None and s[j] != expected:
            break
        if s[j] != ".":
            expected = s[j]
        j += 1
    answer += j - i
print(answer)
```

Но если $s[i:j]$ – неразнообразная, то и $s[i+1:j]$ – также неразнообразная. Поэтому значение j нет необходимости сбрасывать в i на каждой итерации внешнего цикла. Здесь однако надо помнить, что при увеличении i строка, которая содержала какую-то букву, может стать строкой, которая состоит только из точек. Этот момент необходимо не пропустить.

```
s = input()
answer = 0
expected, count = None, 0

j = 0
for i in range(len(s)):
    while j < len(s):
        if s[j] != "." and expected != None and s[j] != expected:
            break
        if s[j] != ".":
            expected, count = s[j], count + 1
        j += 1
    answer += j - i

    if s[i] != ".":
        count -= 1
        if count == 0:
            expected = None

print(answer)
```

Поскольку i и j только растут и никогда не убывают, то сложность данного решения $O(n)$ и оно набирает полный балл.

Среднее арифметическое

Если из каждого элемента последовательности вычесть k , то задача сведется к поиску отрезка из элементов с суммой ноль. Действительно, условие $(s_1+s_2+\dots+s_m)/m=k$, эквивалентно тому, что $(s_1-k)+(s_2-k)+\dots+(s_m-k)=0$.

Пусть элементы исходной последовательности после вычитания k обозначаются как $a_1, a_2, \dots, a_n$. Тот факт, что сумма некоторого отрезка $[l..r]$ равна нулю ($a_l+a_{l+1}+\dots+a_r=0$) означает, что совпадают суммы на префиксах $[1..l-1]$ и $[1..r]$ ($a_1+a_2+\dots+a_{l-1}=a_1+a_2+\dots+a_r$). Это наблюдение позволяет решить задачу на полный балл.

Чтобы проверить существует ли отрезок с суммой ноль, заканчивающийся в

позиции i , можно зная сумму sum на префиксе $[1..i]$ проверить встречалась ли такая сумма раньше на другом суффиксе. Если встречалась, то необходимо знать самое первое вхождение. Если не встречалась, то необходимо сохранить позицию, где первый раз встретилась сумма sum . Данную информацию можно поддерживать в словаре.

```
n, k = [int(x) for x in input().split()]
a = [int(x) - k for x in input().split()]

le, ri = 0, -1
pos = {0: -1}

sum = 0
for i in range(n):
    sum += a[i]
    if sum in pos:
        if i - pos[sum] > ri - le:
            le, ri = pos[sum], i
    else:
        pos[sum] = i

if le > ri:
    print(0)
else:
    print(ri - le, le + 2)
```