

**КРИТЕРИИ И МЕТОДИКА ОЦЕНИВАНИЯ
ВЫПОЛНЕННЫХ ОЛИМПИАДНЫХ ЗАДАНИЙ**

муниципального этапа всероссийской олимпиады школьников по информатике

2022/2023 учебный год

9-11 классы

Олимпиада проводится в один тур, продолжительность тура муниципального этапа составляет 240 минут.

Для знакомства с тестирующей системой проводится пробный тур олимпиады

Муниципальный этап олимпиады проводится с использованием автоматической тестирующей системы для ввода и проверки решений участников, например, Яндекс-контест <https://contest.yandex.ru>, Ejudge <http://ejudge.ru>, тестирующей системы ОЦ «Сириус» <https://uts.sirius.online> и др.

Критерии и методика оценивания.

Для оценивания заданий разработана система тестов

№ п/п	задача	Количество тестов	Балл за тест	Максимальное количество баллов
1.	Автобусные остановки	10	10 За каждый тест	100
2.	Наборы пирожных	25	4	100
3.	Матвей и порталы	25	4	100
4.	Неправильный палиндром	20	5	100

Система оценивания каждой задачи указывается в условии задачи. Если используются общие схемы оценивания в разных задачах, например, для каждой задачи указаны подзадачи и определены зависимости между ними, информация об этом может быть указана в других документах, доступных участникам, – листе с информацией о задачах, правилах олимпиады, памятке участника и т. п.

Во время тура участнику автоматической системой проверки может сообщаться информация о результатах проверки его решения. Возможные формы сообщаемой информации:

- участнику сообщаются баллы, набранные решением, и результат проверки решения на каждом тесте («ОК», «Неправильный ответ», «Превышено максимальное время работы» и т. д.). Сами тесты при этом не сообщаются;
- участнику сообщаются только баллы, набранные решением. Информация о прохождении отдельных тестов не сообщается;
- участнику сообщается информация о прохождении только части тестов. Об этих тестах может сообщаться как детальная информация, так и только количество набранных баллов. О результатах проверки на остальных тестах участнику информация не предоставляется. Типичным вариантом использования такой частичной информации является случай, когда максимальные по ограничениям тесты (предполагающие реализацию наиболее эффективного решения) выделяются в группу, оцениваемую в 30-60% от полного балла, и результат проверки на этой группе является скрытым от участника, а открытым является результат проверки на тестах, не требующих наиболее эффективного решения;
- участнику сообщается только результат проверки на тестах из условия (оцениваемых в 0 баллов).

Допускается сочетание разных форм сообщаемой информации о результатах проверки в разных задачах. Не рекомендуется сообщать информацию о результатах проверки только на тестах из условия во всех задачах, так как на региональном этапе такая форма сообщения результатов не используется.

Каждое задание оценивается из максимального количества баллов, указанного в условии задачи.

При использовании потестовой оценки каждый тест оценивается отдельно указанным в условии задачи числом баллов. Балл участника за задачу равен сумме баллов за тесты.

В условии задачи могут быть указаны характеристики набора тестов, например, доля или суммарный балл тестов, подходящих под некоторые дополнительные ограничения.

При использовании подзадач тесты к задаче разбиваются на группы, каждая группа соответствует одной подзадаче. Для каждой подзадачи устанавливается её «стоимость» в баллах. Участник получает баллы за подзадачу, если все тесты группы для этой подзадачи пройдены.

В условии задачи могут быть указаны дополнительные ограничения на начисление баллов за подзадачу, например, требование прохождения тестов необходимых подзадач.

Информация об этом указана в условии задачи.

Тесты, приведённые в условии задачи в качестве примеров, оцениваются в 0 баллов.

Задача 1. Автобусные остановки (100 баллов)

Имя входного файла: стандартный ввод
 Имя выходного файла: стандартный вывод
 Ограничение по времени: 0.5 секунд
 Ограничение по памяти: 512 мегабайт

Автобусные остановки расположены через каждые K метров от начала улицы, то есть на расстоянии $0, K, 2K, 3K$ и т.д. метров от начала. Света прошла от начала улицы N метров, после чего устала и захотела сесть на автобус. Определите, сколько метров нужно пройти Свете до ближайшей остановки.

Формат входных данных

Программа получает на вход два целых числа K и N , записанных в отдельных строках.
 $1 \leq K \leq 2 \times 10^9$, $1 \leq N \leq 2 \times 10^9$.

Формат выходных данных

Программа должна вывести одно целое число - расстояние до ближайшей остановки.

Система оценки

Решение, правильно работающее только для случаев, когда числа K и N не превосходят 10000, будет оцениваться в 60 баллов.

Пример

стандартный ввод	стандартный вывод
600 2000	200

Замечание

Пояснение к примеру. Остановки расположены на расстоянии $0, 600, 1200, 1800$ и т.д. метров. Света прошла 2000 метров, до ближайшей остановки нужно идти 200 метров.

Пример разбора и решения

Остановки расположены в точках, кратных k , поэтому от последней пройденной остановки Света прошла $n \% k$ метров (здесь $\%$ операция взятия остатка от деления, в языке Pascal это операция `mod`). А до следующей остановки нужно идти $k - n \% k$ метров. Ответом будет минимум из этих величин.

Пример решения.

```
k = int(input())
n = int(input())
print(min(n % k, k - n % k))
```

Задача 2. Наборы пирожных (100 баллов)

Имя входного файла: стандартный ввод
 Имя выходного файла: стандартный вывод
 Ограничение по времени: 0.4 секунд
 Ограничение по памяти: 512 мегабайт

На складе кондитерской фабрики хранятся пирожные двух видов - круассаны и эклеры. Круассанов A штук, а эклеров - B штук. Есть неограниченный запас подарочных коробок, в каждую коробку можно положить только три пирожных. При этом требуется, чтобы в коробке были пирожные обоих видов, то есть в одну коробку можно положить два круассана и один эклер или один круассан и два эклера.

Определите, можно ли упаковать все имеющиеся пирожные в коробки и выведите подходящий способ размещения пирожных по коробкам.

Формат входных данных

Программа получает на вход два целых числа A и B , записанных в отдельных строках.

$$1 \leq A \leq 10^9, 1 \leq B \leq 10^9.$$

Формат выходных данных

Если можно разложить все пирожные по коробкам в соответствии с условием задачи, программа должна вывести два целых числа. Первое число равно количеству коробок, в которых лежит два круассана и один эклер. Второе число равно количеству коробок, в которых лежит один круассан и два эклера.

Если разложить все пирожные по коробкам нужным способом нельзя, программа должна вывести одно число -1.

Система оценки

Решение, правильно работающее только для случаев, когда числа A и B не превосходят 100, будет оцениваться в 60 баллов.

Примеры

стандартный ввод	стандартный вывод
4 5	1 2
5 3	-1

Замечание

В первом примере нужно взять одну коробку с двумя круассанами и одним эклером и две коробки с одним круассаном и двумя эклерами. Всего получится 4 круассана и 5 эклеров.

Пример разбора и решения

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

if (a + b) % 3 != 0:
    print(-1)
else:
    n = (a + b) // 3
    if a >= n and b >= n:
        print(a - n, b - n)
    else:
        print(-1)
```

Задача 3. Матвей и порталы (100 баллов)

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	1 секунда
Ограничение по памяти:	512 мегабайт

Матвей вот-вот опоздает на школьный этап ВсОШ. К счастью, недавно в его городе появились порталы.

Город, в котором живет Матвей, можно представить в виде прямой. Всего в городе успели построить N порталов. Портал с номером i расположен в точке с координатой x_i . Если в текущий момент времени вы находитесь в одной точке с каким-нибудь порталом, то можете всего за одну секунду телепортироваться в любой другой портал вне зависимости от расстояния между ними. А время, требуемое для преодоления расстояния между точками с координатами p и q без использования порталов равно $|p - q|$ секунд. Матвей является влиятельным гражданином, поэтому он может использовать систему порталов любое количество раз.

Изначально Матвей находится в точке s , а точка проведения олимпиады имеет координату e . Помогите Матвею понять, как быстро он может попасть на олимпиаду, ведь каждая секунда на счету.

Формат входных данных

В первой строке входных данных записано одно целое число s — начальное положение Матвея.

Во второй строке записано одно целое число e — место проведения олимпиады.

В третьей строке записано количество порталов N ($2 \leq N \leq 2 \cdot 10^5$).

В каждой из N следующих строк записано целое число x_i — координата портала с номером i . Все числа s , e , x_i по модулю не превосходят 10^8 .

Формат выходных данных

Выведите одно число — минимальное количество секунд, которое потребуется Матвею для того, чтобы добраться до места проведения олимпиады.

Система оценки

Решения, правильно работающие только для случаев, когда N не превосходит 10, будут оцениваться в 20 баллов. Решения, правильно работающие только для случаев, когда N не превосходит 1 000, будут оцениваться в 60 баллов.

Пример

стандартный ввод	стандартный вывод
0	3
4	
3	
1	
3	
5	

Замечание

Рассмотрим пример из условия. Если бы Матвей не мог пользоваться порталами, он бы смог добраться до точки проведения олимпиады за $|0 - 4| = 4$ секунды. Однако, можно действовать так:

1. Дойти до портала с номером 1 за $|0 - 1| = 1$ секунду.

2. Телепортироваться в портал с номером 2 за одну секунду.

3. Дойти от портала с номером 2 до точки проведения олимпиады за $|3 - 4| = 1$ секунду.

Суммарно получаем $1 + 1 + 1 = 3$ секунды.

Пример разбора и решения

Для начала заметим, что всегда можно дойти от изначальной позиции Матвея до места проведения олимпиады, не пользуясь порталами. Данное перемещение можно осуществить за $|s - e|$ секунд. Теперь следует рассмотреть случай, когда выгодно воспользоваться порталами.

Нетрудно понять, что в этом случае оптимальный алгоритм выглядит так:

1. Дойти от изначальной позиции до ближайшего к ней портала.
2. Телепортироваться в ближайший к месту проведения олимпиады портал.
3. Дойти от этого портала до места проведения олимпиады.

Таким образом, единственное, что нужно сделать — это найти ближайшие к изначальной позиции и месту проведения олимпиады порталы. Это можно сделать во время считывания данных. Будем поддерживать две переменные — расстояния до ближайших порталов к двум требуемым точкам. Во время считывания позиции очередного портала нужно проверить, находится он ближе к какой-либо из двух точек, чем ранее найденный портал, или нет. Ниже приведен пример решения задачи на языке Python.

```
s = int(input())
e = int(input())
n = int(input())
near_s = 10 ** 9
near_e = 10 ** 9
for i in range(n):
    x = int(input())
    if abs(x - s) < abs(near_s - s):
        near_s = x
    if abs(x - e) < abs(near_e - e):
        near_e = x
print(min(abs(s - e), abs(s - near_s) + 1 + abs(near_e - e)))
```

Задача 4. Неправильный палиндром (100 баллов)

Имя входного файла: стандартный ввод
 Имя выходного файла: стандартный вывод
 Ограничение по времени: 0.5 секунд
 Ограничение по памяти: 512 мегабайт

Палиндромом называется слово, которое одинаково читается как слева направо, так и справа налево, например, в английском языке такими словами являются «gaga» и «gasesag».

Света изучает английский язык и решила принять участие в дистанционном конкурсе знатоков английского языка. Но, когда она писала ответ на задание «найдите самое длинное слово, которое является палиндромом», ошиблась и нажала на клавиатуре одну лишнюю клавишу.

Определите, какую букву нужно удалить в набранном Светой слове, чтобы это слово стало палиндромом.

Программа получает на вход строку из строчных английских букв, содержащую не менее 2 и не более 100 000 символов.

Программа должна вывести единственное число – номер буквы в строке, при удалении которой слово становится палиндромом. Если при удалении любой буквы слово не станет палиндромом, программа должна вывести число 0.

Примеры входных и выходных данных

Ввод	Вывод
gaseszar	6
car	0

Система оценивания

Решение, правильно работающее только для случаев, когда длина строки не превосходит 100 символов, будет оцениваться в 60 баллов.

Пример разбора и решения

Для начала напишем простой перебор всех возможных ответов: переберём все символы строки, удалим их, если после удаления символа получится палиндром, выведем номер этого символа. Если не нашлось такого символа, выведем 0. Пример решения.

```
s = input()
for i in range(len(s)):
    if s[:i] + s[i + 1:] == (s[:i] + s[i + 1:]][::-1]:
        print(i + 1)
        break
else:
    print(0)
```

В этом решении используется ряд удобств языка Python, поэтому объясним подробнее.

`s[:i]` – это срез строки: подстрока, состоящая из символов до i -го, не включая i -й символ.

`s[i + 1:]` - это срез, от символа с индексом $i + 1$ до конца строки.

`s[:i] + s[i + 1:]` - это конкатенация двух срезов, то есть исходная строка, без символа с индексом i .

`[::-1]` – это срез из символов строки, взятых с шагом -1 , то есть строка, развёрнутая в обратном порядке, то есть в инструкции `if` проверяется, верно ли, что после выбрасывания символа с индексом i получается палиндром. Программа выводит $i + 1$ потому что в языке Python индексация элементов строки начинается с 0, а нужно вывести ответ в нумерации символов с 1.

Инструкция `else` относится к циклу `for`, она выполняется после завершения цикла. Если цикл был прерван по инструкции `break`, то `else` не выполняется, поэтому число 0 будет напечатано, только если не был найден ответ ранее.

Поскольку проверка на палиндромность осуществляется за время, пропорциональное длине строки (нужно развернуть строку и сравнить две строки), то такое решение будет иметь сложность $O(n^2)$, где n – длина строки, и будет набирать 60 баллов.

Приведём решение, имеющее сложность $O(n)$, набирающее полный балл. Сравним первый и последний символ. Если они различаются, то, значит, один из них должен быть удалён. Если они равны, то сравним второй и предпоследний символ, если они равны, то перейдём к следующим от концов строки символам и т.д.

Нам необходимо найти такие два символа, отстоящие на равное расстояние от концов строки, которые различаются. Один из этих символов должен быть удалён. Проверим, что после удаления одного из этих символов, строка становится палиндромом, то есть ответ найден. Если же при удалении каждого из этих двух символов строка не становится палиндромом, то задача не имеет решения и нужно вывести 0.

Наконец, если такие символы не были найдены, то строка уже является палиндромом. В этом случае нужно удалить средний символ в строке (в случае строки чётной длины – один из двух средних символов).

Пример решения.

```
s = input()
i = 0
j = len(s) - 1
while i < j and s[i] == s[j]:
    i += 1 # Поиск двух различных символов i и j,
    j -= 1 # равноудалённых от концов строки
if i >= j: print(i + 1) # строка – уже палиндром, выводим центральный символ
elif s[:i] + s[i + 1:] == (s[:i] + s[i + 1:]>::-1):
    print(i + 1) # можно удалить символ i
elif s[:j] + s[j + 1:] == (s[:j] + s[j + 1:]>::-1):
    print(j + 1) # можно удалить символ j
else:
    print(0) # решения нет
```