

Задания для обучающихся 7-8 классов

Общие рекомендации (ознакомить участников)

Каждому участнику муниципального этапа предлагается для решения пять задач. Решением задачи является исходный текст программы на любом из языков программирования (Pascal, C++, python, Basic). Программа должна вводить исходные данные из текстового файла input.txt (если иное не оговорено в условии задачи) выводить результат в текстовый файл output.txt (если иное не оговорено в условии задачи). Жюри не рассматривает не консольные приложения, в частности программы, использующие формы. В случае, если вы сдаете для оценки несколько вариантов решения задачи, то жюри не будет оценивать ни один из вариантов – вам будет поставлено 0 баллов за решение данной задачи.

Для каждой задачи у жюри имеется 20 тестов. Тесты, приведенные в условиях задачи, могут входить или не входить в набор тестов жюри. За каждый успешно пройденный тест ставится один балл. Тест считается пройденным успешно, если программа вывела верный результат, при этом не вывела никаких лишних символов или сообщений об ошибках и предупреждений. Таким образом каждый участник муниципального этапа может набрать 100 баллов. Даже если вы не уверены в правильности решения, сдайте неполное решение – при прохождении тестов вам могут быть начислены баллы.

Задание А. Цепь

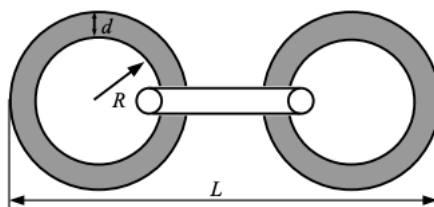
Имя файла: chain

Входной файл: input.txt

Выходной файл: output.txt

Максимальная оценка: 20 баллов

Из проволоки толщиной d миллиметров сделали кольца. Внутренний радиус каждого кольца составляет R миллиметров. Всего сделали n колец и их соединили в цепь. Определите длину получившейся цепи. На рисунке изображен пример для $n = 3$



Вход

Программа получает на вход три числа, записанных в отдельных строках. В первой строке задана толщина проволоки d (в миллиметрах). Во второй строке задан внутренний радиус кольца R (в миллиметрах). В третьей строке задано число звеньев n . Все числа – натуральные, не превосходящие 100, при этом $d < R$.

Выход

Программа должна вывести одно целое число L – суммарную длину получившейся цепи.

Пример входа и выхода:

input.txt	output.txt
2	64
10	
3	

Решение.

Язык программирования Python

```
f = open('input.txt', 'r')
d = int(f.readline())
R = int(f.readline())
n = int(f.readline())
L = 0
```

```
even_link = 2 * R + 2 * d #считаем длину четного звена цепи
odd_link = 2 * R - 2 * d #считаем длину нечетного звена цепи
for i in range(n): #пробегаем каждое звено
    if (i % 2 == 0): #если четное
        L = L + even_link
    else: #если нечетное
        L = L + odd_link
f = open('output.txt', 'w+')
f.write(str(L))
f.close()
```

Критерии оценивания.

Каждый успешно пройденный тест – 1 балл

Задание Б. К-счастливые билеты

Имя файла: klucky

Входной файл: input.txt

Выходной файл: output.txt

Максимальная оценка: 20 баллов

Известно, что билет с шестизначным номером называется счастливым, если сумма первых трёх цифр номера равна сумме последних трёх цифр. В общем случае билет называют К-счастливым, если сумма первых трёх цифр номера в точности равна произведению суммы последних трёх цифр на число К. В этой терминологии «просто счастливые» билеты – это 1-счастливые билеты. Ваша задача – по заданному натуральному числу К найти количество К-счастливых билетов среди всех билетов с номерами от 000000 до 999999.

Вход

Во входном файле записано натуральное число К ($1 \leq K \leq 100$).

Выход

Запишите в выходной файл найденное количество К-счастливых билетов.

Пример входа и выхода:

input.txt	output.txt
1	55252

Решение.

Язык программирования Python

```
f = open('input.txt', 'r')
k = int(f.read())
s = 1 #учитываем 000000

for i in range(1, 1000000):
    a = [int(x) for x in list(str(i))]
    while len(a) < 6: #подолняем лист нулями для получения 6
        значного числа
        a.insert(0, 0)
    if sum(a[:3])==(k * sum(a[3:])): # определяем к-счастливое
        число умножением суммы второй части числа на k
        s = s + 1
f = open('output.txt', 'w+')
f.write(str(s))
f.close()
```

Обратите внимание, что ученик может проверить сначала условие $k \leq 27$ и только после этого начинать переборное решение – это не должно считаться ошибкой.

Критерии оценивания.

Каждый успешно пройденный тест – 1 балл

Задание В. Игра

Имя файла: game

Входной файл: input.txt

Выходной файл: output.txt

Максимальная оценка: 20 баллов

Анна и Борис играют в игру «загадай число». Оба игрока записывают на бумаге число от 1 до 6. Затем один из них бросает кубик, на гранях которого нарисовано, соответственно 1, 2, 3, 4, 5 и 6 точек. Выигрывает тот игрок, чье число оказывается ближе к количеству точек, выпавшему на кубике. Например, если Анна загадала 3, Борис загадал 6, а выпало 4, то выигрывает Анна. Если Анна загадала 2, Борис загадал 4, а выпало 3, то никто не выиграл. Анна и Борис играют несколько раз подряд. Ваша задача – посчитать, сколько раз выиграл каждый из них.

Вход

В первой строке входного файла записано натуральное число N – количество игр ($1 \leq N \leq 1000$). В остальных N строках записана информация об играх. В каждой из этих строк сначала записано число, загаданное Анной, затем число, загаданное Борисом, и затем количество точек, выпавшее на кубике.

Выход

Запишите в выходной файл количество игр, выигранных Анной, и количество игр, выигранных Борисом. Числа необходимо разделять пробелами.

Пример входа и выхода:

input.txt	output.txt
3	1 1
2 6 5	
2 6 1	
2 6 4	

Решение.

Язык программирования Python

```
f = open('input.txt', 'r')
k = int(f.readline())
anna, boris = [0, 0]
for line in f: # считываем каждую строку с игрой
    set_ = [int(x) for x in list(line) if (x != ' ') and (x != '\n')] # создаем массив из трех чисел: 1 - анна, 2 - борис, 3 - кубик
    if abs(set_[0] - set_[2]) > abs(set_[1] - set_[2]): # если
```

```
анна ближе - увеличиваем ее счет
    anna = anna + 1
    elif abs(set_[0] - set_[2]) < abs(set_[1] - set_[2]):
        boris = boris + 1 #если борис ближе - увеличиваем его
счет
f = open('output.txt', 'w+')
f.write(str(anna) + ' ' + str(boris))
f.close()
```

Обратите внимание, что условие обязательно должно проверяться дважды (сначала на больше и затем на меньше или наоборот). Если ученик проверит одно условие, а второе «унесет» в ветку «иначе», то это будет ошибкой, так как по условию задачи если игроки получили одинаковое расстояние до выпавшего числа, то никому не начисляется балл.

Критерии оценивания.

Каждый успешно пройденный тест – 1 балл

Задание Г. Шифровка

Имя файла: encoding

Входной файл: input.txt

Выходной файл: output.txt

Максимальная оценка: 20 баллов

Петя и Вася обмениваются зашифрованными сообщениями. Они используют такой алгоритм шифрования. Берут секретное число K , известное на всём свете только им двоим, и в каждом слове сообщения K раз первую букву переставляют в конец слова. Например, если $K=3$, то слово «ШКОЛА» превратится сначала в «КОЛАШ», затем в «ОЛАШК» и, наконец, в «ЛАШКО». После этого полученное слово записывают в обратном порядке: «ОКШАЛ». Такое шифрование – довольно тяжёлый труд, помогите Пете и Васе его автоматизировать.

Вход

В первой строке входного файла записано число K ($0 \leq K \leq 1,000,000,000$). Во второй строке записано одно русское слово. Длина слова не превосходит 100 букв. Все буквы в слове заглавные.

Выход

Запишите в выходной файл зашифрованное слово.

Пример входа и выхода:

input.txt	output.txt
3 ШКОЛА	ОКШАЛ

Решение.

Язык программирования Python

```
f = open('input.txt', 'r')
k = int(f.readline())
word = f.readline()
for i in range(k):
    word = word[1:] + word[0] #переставляем первую букву в конец
    слова ровно k раз
f = open('output.txt', 'w+')
f.write(word[::-1]) #выводим слово, переворачивая его
f.close()
```

Критерии оценивания.

Каждый успешно пройденный тест – 1 балл

Задание Д. Манхэттен

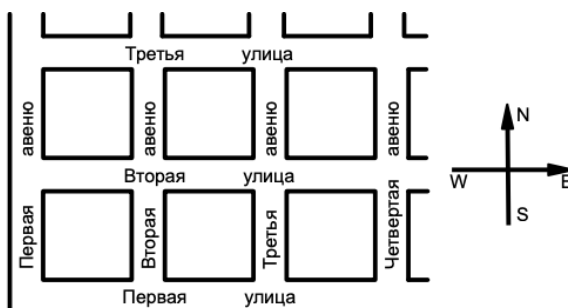
Имя файла: manhattan

Входной файл: input.txt

Выходной файл: output.txt

Максимальная оценка: 20 баллов

Кварталы Манхэттена состоят из авеню, направленных с юга на север и улиц, направленных с запада на восток. Все улицы и авеню пронумерованы числами, начиная с 1 подряд (первая улица, вторая улица, третья улица и т. д.). Передвигаться можно только по улицами или по авеню.



Миша впервые попал на Манхэттен. Сейчас он стоит на пересечении авеню номер x_1 и улицы номер y_1 . Ему нужно попасть на перекресток авеню номер x_2 и улицы номер y_2 . Определите маршрут, который он должен пройти.

Вход

Программа получает на вход 4 числа: x_1 , y_1 , x_2 , y_2 , записанных в отдельных строках. Все числа – натуральные, не превышают 10^3 . Начальное и конечное расположение Миши не совпадают.

Выход

Программа должна вывести последовательность из латинских заглавных букв, описывающих маршрут, которому должен следовать Миша. Буква «N» обозначает перемещение на один квартал на север, «S» — на юг, «W» — на запад, «E» — на восток. Программа должна вывести самый короткий из всех возможных маршрутов, если же кратчайших маршрутов существует несколько, то программа должна вывести любой из них (но только один).

Пример входа и выхода:

input.txt	output.txt
1	EEESS
3	
4	

1	
---	--

Решение.

Язык программирования Python

```
f = open('input.txt', 'r')
x1 = int(f.readline())
y1 = int(f.readline())
x2 = int(f.readline())
y2 = int(f.readline())
f = open('output.txt', 'w+')
f.write("E"*(x2-x1) + "W"*(x1-x2) + "N"*(y2-y1) + "S"*(y1-y2))
f.close()
```

Если $x_2 > x_1$, то Миша должен переместиться на $x_2 - x_1$ квартала на восток, поэтому программа должна вывести $x_2 - x_1$ символ «E», что можно сделать при помощи цикла. Иначе Миша должен переместиться на $x_1 - x_2$ квартала на запад, поэтому программа должна вывести $x_1 - x_2$ символ «W». Аналогично рассматривается передвижение на север и на юг. Поэтому, решение задачи через циклы может выглядеть так

```
...
if x2 > x1:
    for i in range(x2 - x1):
        f.write("E")
else:
    for i in range(x1 - x2):
        f.write("W")
if y2 > y1:
    for i in range(y2 - y1):
        f.write("N")
else:
    for i in range(y1 - y2):
        f.write("S")
...
```

Критерии оценивания.

Каждый успешно пройденный тест – 1 балл