Задания для обучающихся 9-11 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 120 баллов. Даже если вы не уверены в правильности решения, сдайте неполное решение – при прохождении тестов вам могут быть начислены баллы.

Задание А. Цифры

Имя файла: digits

Входной файл: input.txt

Выходной файл: output.txt

Максимальная оценка: 20 баллов

Запишем последовательно и без пробелов натуральные числа от 1 до N. Например, для N = 13 мы получим последовательность 12345678910111213. А теперь посчитаем, сколько раз каждая цифра входит в эту последовательность. Цифра 0 входит 1 раз, цифра 1 – 6 раз, цифра 2 – 2 раза и так далее. Напишите программу, которая для заданного числа N будет находить количество вхождений в такую последовательность для всех цифр от 0 до 9.

Вход

Во входном файле записано натуральное число N (1 <= N <= 10000)

Выход

Запишите в выходной файл через пробел десять чисел – количество вхождений цифр 0, 1, …, 9 в получившуюся последовательность.

Пример входа и выхода:

input.txt	output.txt
3	0 1 1 1 0 0 0 0 0 0
13	1 6 2 2 1 1 1 1 1 1

Решение.

Язык программирования Python

```
f = open('input.txt', 'r')
n = int(f.readline())
sequence = "".join([str(x) for x in range(1, n + 1)]) #создаем
строку из чисел от 1 до N
f = open('output.txt', 'w+')
for i in range(0, 10): # для каждой цифры 0, 1, 2 ... 9
    if i == 9:
        f.write(str(sequence.count(str(i)))) #если считаем 9 то
не надо ставить пробел в конце
    else:
        f.write(str(sequence.count(str(i))) + ' ') #считаем
количество цифр в строке
f.close()
```

Критерии оценивания.

Каждый успешно пройденный тест – 1 балл

Задание Б. Олигархи

Имя файла: olig

Входной файл: input.txt

Выходной файл: output.txt

Максимальная оценка: 20 баллов

Собрались как-то олигархи и стали хвастаться – у кого денег больше. Первый олигарх говорит – у меня одних долларов 100 миллиардов, а ещё евров 50 миллиардов, и рублей полно. Второй ему отвечает – подумаешь, кому сейчас твои доллары нужны, у меня вот британских фунтов стерлингов 76 миллиардов с копейками, да тугрики монгольские, да динары алжирские... Тут все олигархи как закричат, как ногами затопают… и подрались, синяков друг другу наставили, а одному олигарху зуб выбили. А всё потому, что олигархи были глупые – надо было не кричать, и не драться, а спокойно пересчитать все их сбережения, непосильным трудом нажитые, в родные русские рубли, тогда сразу бы ясно стало, кто из них самый богатый. Напишите программу, которая определяет самого богатого из олигархов.

Вход

В первой строке входного файла записаны натуральные числа **N** – количество олигархов и **M** - количество разных валют, в которых они хранят свои сбережения (2 <= **N** <= 1000, 1 <= **M** <= 100). Во второй строке записано **M** вещественных чисел – курсы валют по отношению к рублю. В остальных **N** строках записано по **M** целых чисел в каждой. **j**-е число в **i**-ой из этих строк равно количеству миллиардов в **j**-ой валюте у **i**-го олигарха. Валюта принимает значение от 1 до 100. Количество миллиардов приминает значение от 1 до 100000

Выход

Запишите в выходной файл номер самого богатого олигарха. Если таких олигархов несколько, запишите наименьший из номеров.

Пример входа и выхода:

input.txt	output.txt
2 1 1.0 100 101	2
3 4 1.0 36.350 25.943 48.021	3

100 15 25 0 200 40 0 0 150 0 30 25	

Решение.

Язык программирования Python

```
f = open('input.txt', 'r')
N, M = [int(x) for x in f.readline().split(' ') if x != '\n']
currency = [float(x) for x in f.readline().split(' ') if x !=
'\n']
moneyInRubles = [0 for i in range(N)]
for i in range(0, N):
    money = [float(x) for x in f.readline().split(' ') if x !=
'\n'] #считываем сбережения N-го олигарха
    moneyInRubles[i] = sum([a*b for a,b in zip(currency,
money)]) #записываем в список сбережение
f = open('output.txt', 'w+')
f.write(str(moneyInRubles.index(max(moneyInRubles)) +
1))#находим максимальное сбережение и возвращаем его индекс
f.close()
```

Критерии оценивания.

Каждый успешно пройденный тест – 1 балл

Задание B. Соревнование картингистов

Имя файла: race

Входной файл: input.txt

Выходной файл: output.txt

Максимальная оценка: 20 баллов

После очередного этапа чемпионата мира по кольцевым автогонкам на автомобилях с открытыми колесами Формула-А гонщики собрались вместе в кафе, чтобы обсудить полученные результаты. Они вспомнили, что в молодости соревновались не на больших болидах, а на картах – спортивных автомобилях меньших размеров. Друзья решили выяснить победителя в одной из гонок на картах. Победителем гонки являлся тот гонщик, у которого суммарное время прохождения всех кругов трассы было минимальным. Поскольку окончательные результаты не сохранились, то каждый из n участников той гонки вспомнил и выписал результаты прохождения каждого из m кругов трассы. К сожалению, по этой информации гонщикам было сложно вычислить победителя той гонки. В связи с этим они попросили сделать это вас.

Требуется написать программу, которая вычислит победителя гонки на картах, о которой говорили гонщики.

Вход

Первая строка входного файла содержит два целых числа n и m ($1 \le n, m \le 100$). Последующие $2 \cdot n$ строк описывают прохождение трассы каждым из участников. Описание прохождения трассы участником состоит из двух строк. Первая строка содержит имя участника с использованием только латинских букв (строчных и заглавных). Имена всех участников различны, строчные и заглавные буквы в именах различаются. Вторая строка содержит m положительных целых чисел, где каждое число – это время прохождения данным участником каждого из m кругов трассы (каждое из этих чисел не превосходит 1000). Длина каждой строки не превышает 255 символов.

Выход

В выходной файл необходимо вывести имя победителя гонки на картах. Если победителей несколько, требуется вывести имя любого из них.

Пример входа и выхода:

input.txt	output.txt	output.txt
5 3 Sumaher 2 1 1 Barikelo 2 1 2 Olonso 1 2 1 Vasya 1 1 1 Fedya 1 1 1	Fedya	Vasya

Решение.

Язык программирования Python

```
f = open('input.txt', 'r')
N, M = [int(x) for x in f.readline().split() if x != '\n']
times = {} #время каждого участник в виде словаря Имя - время
for i in range(N):
    name = f.readline().rstrip('\n') #имя участника
    times[name] = sum([int(x) for x in f.readline().split() if x
!= '\n']) #записываем в словарь значение времени
соответствующему имени участника
minTime = min(times.values()) #минимальное время среди всех
участников
winners = [key for key in times if times[key] == minTime]
#создаем список победителей
f = open('output.txt', 'w+')
f.write(winners[0]) #выводим первого в списке
f.close()
```

Для проверяющего. Обратите внимание, что в задаче необходимо вывести имя любого участника-победителя. В наборе тестов в соответствующем входному тесту выходном файле будут перечислены все победители. Но, у ученика в выходном файле должно быть ТОЛЬКО ОДНО имя, любое, которое будет в выходном файле набора теста.

Критерии оценивания.

Каждый успешно пройденный тест – 1 балл

Задание Г. Автомобильные номера

Имя файла: cars

Входной файл: input.txt

Выходной файл: output.txt

Максимальная оценка: 20 баллов

В Российской Федерации на разных видах транспортных средств устанавливаются разные по формату регистрационные знаки («автомобильные номера»). Вот пример нескольких возможных форматов регистрационных знаков.

№	Пример	Описание формата	Тип транспортного средства
1	Y019KM	Буква, три цифры, две буквы	Частные транспортные средства
2	AB179	Две буквы, три цифры	Общественный транспорт и такси
3	OH2645	Две буквы, четыре цифры	Прицепы
4	3384CT	Четыре цифры, две буквы	Мотоциклы

В этой задаче «буквой» может быть любая заглавная буква латинского алфавита. Напишите программу, которые по регистрационному знаку определяет его тип или определяет, что регистрационный знак некорректен (выводится 0).

Вход

Программа получает на вход три строки текста, каждая строка содержит один образец регистрационного знака (возможно, некорректный). Каждый образец содержит от 1 до 10 символов, являющихся цифрами и заглавными латинскими буквами (других символов во входных данных быть не может).

Выход

Программа должна вывести для каждого образца число, соответствующее типу транспортного средства, как в приведенной таблице, то есть 1 — для частных транспортных средств, 2 — для общественного транспорта, 3 — для прицепов, 4 — для мотоциклов. Если номерной знак некорректен (не подходит ни к одному из указанных типов), то необходимо вывести число 0. Каждое число необходимо выводить в отдельной строке.

Пример входа и выхода:

input.txt	output.txt
Y019KM A9999 OH2645	1 0 3

Решение.

Язык программирования Python

```
f = open('input.txt', 'r')
numbers = f.read().splitlines()
f = open('output.txt', 'w+')
for number in numbers:
    C = ''
    for char in number: #создаем маску номера
        if '0' <= char <= '9':
            C += '0'
        else:
            C += 'A'
    #проверяем маску и присваиваем соответствующий тип
    if C == 'A000AA':
        f.write(str(1) + '\n')
    elif C == 'AA000':
        f.write(str(2) + '\n')
    elif C == 'AA0000':
        f.write(str(3) + '\n')
    elif C == '0000AA':
        f.write(str(4) + '\n')
    else:
        f.write(str(0) + '\n')
f.close()
```

Критерии оценивания.

Каждый успешно пройденный тест – 1 балл

Задание Д. Города

Имя файла: countries

Входной файл: input.txt

Выходной файл: output.txt

Максимальная оценка: 20 баллов

Юный программист решил придумать собственную игру. Игра происходит на поле размером N × N клеток, в некоторых клетках которого расположены города (каждый город занимает одну клетку; в каждой клетке может располагаться не более одного города). Всего должно быть чётное количество городов. Изначально про каждую клетку игрового поля известно, расположен ли в ней город или нет. Чтобы начать игру, необходимо разделить игровое поле на два государства так, чтобы в каждом государстве было поровну клеток-городов. Граница между государствами должна проходить по границам клеток таким образом, чтобы из любой клетки каждого государства существовал путь по клеткам этого же государства в любую другую его клетку (из клетки можно перейти в соседнюю, если они имеют общую сторону). Каждая клетка игрового поля должна принадлежать только одному из двух государств, при этом государства не обязаны состоять из одинакового количества клеток. Требуется написать программу, которая с учетом сказанного разделит клетки заданного игрового поля между двумя государствами.

Вход

Первая строка входного файла содержит одно целое положительное число N, задающее размер игрового поля ($1 \le N \le 50$). Последующие N строк содержат по N заглавных латинских букв (без пробелов), кодирующих соответствующие клетки игрового поля: ‘C’ обозначает клетку, занятую городом, ‘D’ – пустую клетку. Гарантируется, что на поле есть хотя бы два города и всего их четное число.

Выход

Выходной файл должен содержать N строк по N цифр (без пробелов) в каждой, кодирующих соответствующие клетки. Цифра 1 обозначает, что данная клетка принадлежит первому государству, цифра 2 – данная клетка принадлежит второму государству. Если решений несколько, необходимо вывести любое из них.

Пример входа и выхода:

input.txt	output.txt
3	111

DDD DDC DDC	112 222
5 DDDDD CDCDC DCCDC DDDDD DDDDD	11111 11112 22222 22222 22222

Решение.

Язык программирования Python

```
f = open('input.txt', 'r')
n = int(f.readline())
towns = 0 #количество городов
territory = [[] for x in range(n)]
for i in range(n):
    territory[i] = [x for x in f.readline() if x != '\n']
#заносим каждую строку в список для последующего разделения на
государства
    towns += territory[i].count('C')
townsPerCountry = towns // 2 #количество городов, которые должны
быть в каждом государстве

addedTowns = 0 #Количество городов отнесенных к государству
#проходим по каждой клетке-территории и считаем встречающийся
город и помечаем
#каждую пройденную клетку государством 1, как только прошли
необходимое количество
#городов остальные клетки будут государством 2. Таким образом
гарантируется связность
#городов и государств
for i in range(n):
    for j in range(len(territory[i])):
        if ((territory[i][j] == 'C')):
            addedTowns += 1
            if (addedTowns < townsPerCountry):
                territory[i][j] = '1'
            else:
                territory[i][j] = '2'
        else:
            if (addedTowns < townsPerCountry):
                territory[i][j] = '1'
            else:
                territory[i][j] = '2'
f = open('output.txt', 'w+')
for i in territory:
    f.write(''.join(i) + '\n')
f.close()
```

Для проверяющего. Обратите внимание, на то, что ученик может сначала отмечать территорию государства 2, а потом 1 – это не должно считаться ошибкой. При этом, стоит учесть, что решение может иметь другой вид, например в тестовом наборе 2 из задачи вместо

5 DDDDD CDCDC DCCDC DDDDD DDDDD	11111 11112 22222 22222 22222

Может быть

5 DDDDD CDCDC DCCDC DDDDD DDDDD	11111 12221 12221 11111 11111

Главное, чтобы территория государства 1 была доступна из 2 и наоборот. То есть недопустимо, чтобы алгоритм ученика «бил» территорию государств. Например так

5 DDDDD CDCDC DCCDC DDDDD DDDDD	11111 11112 22222 22222 11111

Критерии оценивания.

Каждый успешно пройденный тест – 1 балл

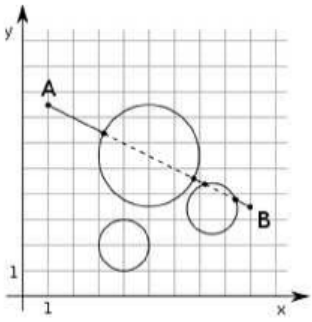
Задание E. Города

Имя файла: traps

Входной файл: input.txt

Выходной файл: output.txt

Максимальная оценка: 20 баллов



Штуша-Кутуша охотился на Фантика. С этой целью он вырыл ряд круглых непересекающихся ям-ловушек. Однако Фантик был очень внимательным и спортивным слонёнком, поэтому двигаясь по прямолинейной траектории из точки A в точку B, подойдя к самому краю ямы-ловушки, он перепрыгивал её точно до следующего края, не отклоняясь от выбранной траектории. Требуется написать программу, определяющую какое расстояние «напрыгал» (провёл в прыжке) Фантик с точностью до второго десятичного знака.

Вход

В первой строке вещественные числа x_A, y_A, x_B, y_B ($-10000 \le x_A, y_A, x_B, y_B \le 10000$) – координаты точек A и B. Во второй строке число N ($1 \le N \le 32767$) количество ям-ловушек Штуши-Кутуши. Далее N строк по 3 вещественных числа – x_k, y_k – координаты центра ямы- ловушки, R_k – её радиус ($-10000 \le x_k, y_k, R_k \le 10000$, $1 \le k \le N$).

Выход

Одно число, с округлением до двух десятичных знаков, – «напрыганное» Фантиком расстояние.

Пример входа и выхода:

input.txt	output.txt
1 7.5 9 3.5 3 4 2 1 5 5.5 2 7.5 3.5 1	5.48
1 5 2 5 1 4 4 1	0.00

Решение.

Язык программирования Python

```
def intersect(x1, y1, x2, y2, x0, y0, r): #функция определения
пересечений прямой с окружностью
    #найдем коэффициенты квадратного уравнения
    a = x2*x2 - 2*x2*x1 + x1*x1 + y2*y2 - 2*y2*y1 + y1*y1
    b = 2*(x1*x2 - x1*x1 - x0*x2 + x0*x1) + 2*(y1*y2 - y1*y1 -
y0*y2 + y0*y1)
    c = x1*x1 - 2*x1*x0 + x0*x0 + y1*y1 - 2*y1*y0 + y0*y0 - r*r
    d = b*b - 4*a*c
    if d >= 0: #если квадратное уравнение имеет решение
        t1 = (-b + d**0.5) / (2*a) #находим корни
        t2 = (-b - d**0.5) / (2*a)
        #возвращаем список из 4 точек пересечений прямой с
окружностями
        return [x1*(1 - t1) + x2*t1, y1*(1 - t1) + y2*t1, x1*(1
- t2) + x2*t2, y1*(1 - t2) + y2*t2]
    else:
        #возвращаем пустой список
        return []
f = open('input.txt', 'r')
Ax, Ay, Bx, By = [float(x) for x in f.readline().split() if x !=
'\n']
n = int(f.readline())
s = 0 #длина прыжков зайца

for i in range(n):
    x, y, r = [float(x) for x in f.readline().split() if x !=
'\n']
    intersectResult = intersect(Ax, Ay, Bx, By, x, y, r)
    if intersectResult: #если список не пустой
        s += ((intersectResult[0] - intersectResult[2])**2 +
(intersectResult[1] - intersectResult[3])**2)**0.5
f = open('output.txt', 'w+')
f.write('%.2f' % s)
f.close()
```

Решение задачи сводится к нахождению точек пересечения прямой с каждой окружностью. При нахождении точек пересечения вычисляется расстояние между ними и складывается в переменную-сумматор расстояний точек пересечений. В функции intersect используются формулы, выведенные из уравнения окружности и уравнения прямой, заданной параметрически.

Критерии оценивания.

Каждый успешно пройденный тест – 1 балл