

Рябцев Роман Денисович

Класс 8

Статус: Призер!

Набрано баллов: 19

Задание №: 1

Распознавание лиц

Ограничение по памяти: 256 мб

Ограничение по времени: 1 сек

[Скачать NotePad++](#)

Условие:

Не всегда задача распознавания лица сводится к определению того, как расположены его части. Иногда необходимо ответить на вопрос - куда повернуто лицо. Например, таким образом можно сказать в какую сторону смотрит человек или общаются ли два человека между собой.

Если взять некоторые точки на лице человека (например, глаза, уголки губ, нос), то приняв их за вершины, можно построить многогранник. (см. рис. 1)

Для простоты, допустим, что многогранник - пирамида, в основании которой - квадрат.

Определите время, за которое будут обслужены все клиенты.

Если лицо направлено в камеру, то основание пирамиды лежит в плоскости, параллельной фокальной плоскости камеры (параллельной плоскости светочувствительной матрицы). Очевидно, что проекция основания пирамиды на кадр - квадрат, а вершина пирамиды будет лежать на пересечении диагоналей квадрата. Тогда, если человек повернет свое лицо, например, вверх, то проекция пирамиды на кадр изменится: квадрат может превратиться в просто многоугольник, а вершина пирамиды больше не будет лежать на диагоналях многоугольника. (см. рис. 2)

Ваша задача составить алгоритм, который позволит по проекции правильной пирамиды на кадр определить на какой угол она была повернута.

Входные данные:

В первой строке находится натуральное число h - высота правильной пирамиды в миллиметрах ($1 \leq h \leq 10^9$).

В последующих 4 строках задаются пары вещественных чисел: координаты точек вершин пирамиды, лежащих в основании ($0 \leq x_i, y_i \leq 10^9$).

В последней строке задаётся пара вещественных чисел: координаты точки вершины пирамиды, не лежащей в основании.

Выходные данные:

В выходной файл должно быть выведено единственное значение - угол поворота пирамиды относительно оси, проходившей до поворота через вершину пирамиды и центр квадрата в основании. Угол укажите в радианах с точностью до сотых.

Пример:

Входные данные	Выходные данные
230	0.26
84 371	
85 688	
398 697	
397 380	
191 567	

Рис.1

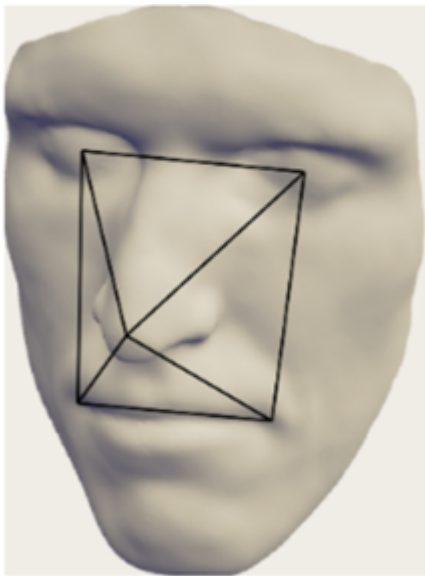
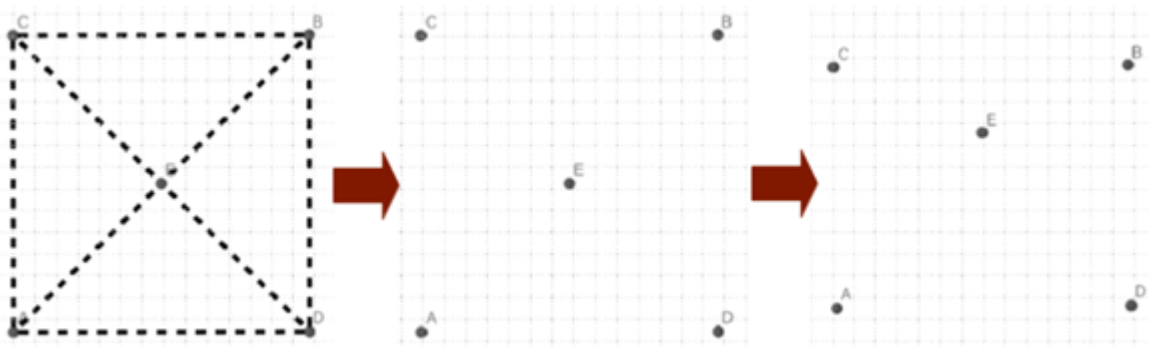


Рис.2



Напишите программу и загрузите в формате txt. Вы можете использовать Блокнот или NotePad++

Общий балл за задание: 0

Задание №: 2

Конь-сапер

Ограничение по памяти: 256 мб

Ограничение по времени: 3 сек

[Скачать NotePad++](#)

Условие:

Biston DIynamics недавно представила робота-сапёра HORSEMAN, который может помочь спасти многие жизни. Он

может по внешнему виду схемы контроллера взрывного устройства определить, что это за устройство, и обезвредить его. Но, как и у любой системы, у этого робота есть слабые места. За счёт конструкции его ног он вынужден ходить подобно шахматному коню (за счёт этого он и получил своё название), но испытания показали весьма высокую эффективность данной конструкции (правда, при определенных условиях). Ваша задача написать модуль полного обхода помещения площадью $n*m$ с целью анализа на наличие взрывных устройств (робот должен проанализировать КАЖДЫЙ сектор площадью $1*1$ данного помещения).

Входные данные:

В первой строке находятся 2 целых числа: n и m - ширина и длина помещения ($4 \leq m, n \leq 26$)

Выходные данные:

В выходной файл должно быть выведен путь данного робота. Если он не может полностью обойти данное помещение, то необходимо вывести лишь -1.

Пример:

Входные данные	Выходные данные
4 5	11
	3 2
	4 4
	2 5
	1 3
	2 1
	4 2
	3 4
	1 5
	2 3
	3 1
	1 2
	2 4
	4 5
	3 3
	4 1
	2 2
	1 4
	3 5
	4 2

Напишите программу и загрузите в формате txt. Вы можете использовать Блокнот или NotePad++

Общий балл за задание: 0

Задание №: 3

История любви

Ограничение по памяти: 256 мб

Ограничение по времени: 1 сек

[Скачать NotePad++](#)

Условие:

Петя очень любит строки одинаковой длины.

Он просто без ума от этих прекрасных строк.

Но недавно к ним в класс перевелась девочка Маша, которая любит строки, отличающиеся друг от друга лишь на t символов.

И вот, чтобы понравиться Маше, Петя решил пойти на хитрость и написать программу, которая для двух строк одинаковой длины генерирует строку, которая отличается от каждой из исходных лишь на t символов.

Входные данные:

В первой строке записаны 2 числа n и t ($1 \leq n \leq 10^5$, $0 \leq t \leq n$), где n - длина строк, t - количество символов, на которые результирующая строка должна отличаться от каждой из исходных

Во второй и третьей строке записаны строки длины n , состоящие из строчных английских букв.

Выходные данные:

Выведите строку длины n , отличающуюся от исходных ровно в t символах. Ваша строка должна состоять только из строчных букв английского алфавита. Если же такой строки не существует, выведите -1.

Пример:

Входные данные	Выходные данные
3 3 <u>yva</u> <u>yva</u>	<u>aab</u>

Напишите программу и загрузите в формате txt. Вы можете использовать Блокнот или NotePad++

Общий балл за задание: 0

Работа участника:

```
#input
#values
n = int(input('n = '))
t = int(input('t = '))
#strings
a = input('line1 - ')
b = input('line2 - ')
#=====
def str_cut(val, t):
if t == 0:
return ""
return ".join(list(val)[:t])
#=====
if not t>n:
out = a + str_cut(b, t)
else:
out = -1
print(out)
input('Programm end')
```

Задание №: 4

История любви

Ограничение по памяти: 64 мб

Ограничение по времени: 2 сек

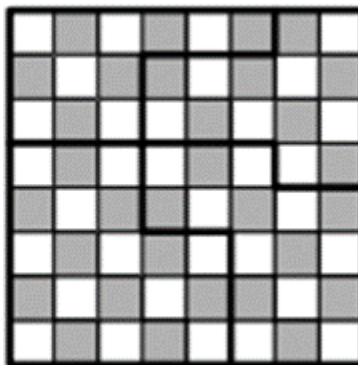
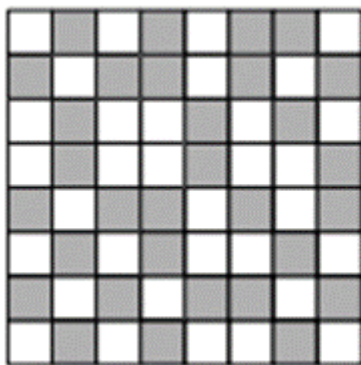
[Скачать NotePad++](#)

Условие:

После успешного завершения строительства двое организаторов олимпиады решили поиграть в шахматы. Но единственная шахматная доска, которую им удалось найти, выглядела весьма странно: как выяснилось, она была склеена из нескольких кусков, которые были вырезаны из правильной шахматной доски.

Конечно, на такой доске играть невозможно. Но организаторов теперь волнует другой вопрос: из какого минимального количества кусков шахматной доски могла быть склеена эта доска?

Например, доска слева может быть склеена из четырёх кусков, как показано справа.



Формат входных данных:

Входной файл содержит восемь строк, состоящих из восьми символов «W» и «B», не разделенных пробелами. Символ «W» обозначает белую клетку, а символ «B» — черную.

Формат выходных данных:

Выведите в выходной файл одно число — минимальное количество кусков, из которых могла быть склеена заданная во входном файле доска.

Пример:

входные данные	выходные данные
WBWBWBBW BWBBWBWB WBWWWBWBW WBWWWBWWB BWBBWBWB WBWBWWBW BWBWBBWB WBWBWWBW	4

Напишите программу и загрузите в формате txt. Вы можете использовать Блокнот или NotePad++

Общий балл за задание: 19

Работа участника:

#ввод данных из файла field.txt ,который должен лежать

#в одной папке с этим файлом

#-----

```
def recognize(text):
```

```
    out = text.split('\n')
```

```
    for i in range(len(out)):
```

```
        out[i] = list(out[i])
```

```
    return out
```

```
def rewrite_the_dop(put, kill):
```

```
    global dop
```

```
    for y in range(len(dop)):
```

```
        for x in range(len(dop[y])):
```

```
            if dop[y][x] == kill:
```

```
                dop[y][x] = put
```

```
def numb_of_fields():
```

```
    global dop
```

```
    val = []
```

```
    for y in range(len(dop)):
```

```
        for x in range(len(dop[y])):
```

```
            b = True
```

```
            for i in val:
```

```
                if dop[y][x] == i:
```

```
                    b = False
```

```
            break
```

```
            if b:
```

```
                val.append(dop[y][x])
```

```
    return len(val)
```

#-----

```
file = open('field.txt','r')
```

```
table = recognize(file.read())
```

```
file.close()
```

```
dop = [[0,0,0,0,0,0,0,0],
```

```
[0,0,0,0,0,0,0,0],
```

```
[0,0,0,0,0,0,0,0],
```

```
[0,0,0,0,0,0,0,0],
```

```
[0,0,0,0,0,0,0,0],
```

```
[0,0,0,0,0,0,0,0],
```

```
[0,0,0,0,0,0,0,0],
```

```
[0,0,0,0,0,0,0,0]]
```

```
#=====
```

```
a = 0
```

```
for y in range(len(table)):
```

```
for x in range(len(table[y])):
```

```
br = False
```

```
if x == 0:
```

```
left = False
```

```
else:
```

```
left = True
```

```
if y == 0:
```

```
up = False
```

```
else:
```

```
up = True
```

```
if up and left:
```

```
if table[y-1][x] != table[y][x] and table[y][x-1] != table[y][x]:
```

```
rewrite_the_dop(dop[y-1][x],dop[y][x-1])
```

```
#a = a - 1
```

```
br = True
```

```
if up:
```

```
if table[y-1][x] != table[y][x]:
```

```
dop[y][x] = dop[y-1][x]
```

```
br = True
```

```
if left:
```

```
if table[y][x-1] != table[y][x]:
```

```
dop[y][x] = dop[y][x-1]
```

```
br = True
```

```
if not br:
```

```
dop[y][x] = a + 1
```

```
a = dop[y][x]
```

```
print(numb_of_fields())
```

Задание №: 5

Поезда

Ограничение по времени: 2 сек

[Скачать NotePad++](#)

Условие:

На железнодорожной ветке руководство железной дороги решило изменить график движения поездов. Тщательный анализ состояния железнодорожного полотна показал, что оптимальным является следующий график движения поездов с учетом остановок на станциях: сначала поезд идет на протяжении T_1 минут со скоростью V_1 метров в минуту, затем T_2 минут со скоростью V_2 метров в минуту, ..., наконец, T_N минут со скоростью V_N метров в минуту. В течение некоторых интервалов поезд может стоять (скорость равна 0).

По действующей инструкции обеспечения безопасности движения поездов расстояние между локомотивами двух следующих друг за другом поездов должно быть не менее L метров. Определите минимально допустимый интервал в минутах между отправлениями поездов, позволяющий им двигаться по этому графику без опасного сближения.

Формат входных данных:

В первой строке входного файла содержится два натуральных числа, задающие минимально допустимое расстояние L и количество участков пути N ($100 \leq L \leq 1000$, $1 \leq N \leq 1000$). Далее следует N пар целых чисел T_i и V_i , задающих график движения поездов ($1 \leq T_i \leq 1000$, $0 \leq V_i \leq 1000$).

Формат выходных данных:

В выходной файл необходимо вывести искомый интервал между отправлениями поездов в минутах, не менее чем с тремя верными знаками после десятичной точки.

Пример:

входные данные	выходные данные
1000 4 10 0 30 80 15 0 20 100	27.500

Напишите программу и загрузите в формате txt. Вы можете использовать Блокнот или NotePad++

Общий балл за задание: 0

