

Практическое занятие №6

Тема: Составление программ со списками в IDE PyCharm

Цель: закрепить усвоенные знания, понятия, алгоритмы, основные принципы составления программ, приобрести навыки составления программ со списками в IDE PyCharm Community

Задание №1

Постановка задачи.

Дан список ненулевых целых чисел размера N. Проверить, чередуются ли в нем положительные и отрицательные числа. Если чередуются, то вывести 0, если нет то вывести порядковый номер 1 элемента, нарушающего закономерность

Тип алгоритма : циклический с ветвлением

Текст программы:

```
# Дан список ненулевых целых чисел размера N. Проверить, чередуются ли в нем
# положительные и отрицательные числа. Если чередуются, то вывести 0, если
# нет
# то вывести порядковый номер 1 элемента, нарушающего закономерность

import random

# инициализация переменных
tf = False
l = []
b = 0
i = 0
r = int(input())

try:
    # создание массива
    while i < r:
        a = random.randint(-3,3)
        i+=1
        if a == 0:
            i = i-1
            continue
        l.append(a)

    print(l)
    s=0

    #Проверка
    for i in l:
```

```

        if i*b > 0:
            tf = True
            print(s-1, 'порядковый номер последнего элемента')
            break
        s += 1

        b = i
    if tf == False:
        print(0)
except:
    print('error')

```

Протокол работы:

```

C:\Projects_1sem_Slusarev\venv\Scripts\python.exe C:\Projects_1sem_Slusarev\PZ_6\PZ_6_1.py
5
[-3, 2, -3, -2, 3]
2 порядковый номер последнего элемента

Process finished with exit code 0

```

Задание 2

Постановка задачи.

Дан целочисленный список A размера N (< 15). Переписать в новый целочисленный список B все элементы с нечетными порядковыми номерами (1,3,...) и вывести размер полученного списка B и его содержимое. Условный оператор не использовать

Тип алгоритма : с ветвлением

Текст программы:

```

#2. Дан целочисленный список A размера N (< 15). Переписать в новый
целочисленный
# список B все элементы с нечетными порядковыми номерами (1,3,...)
# и вывести размер полученного списка B и его содержимое.
# Условный оператор не использовать

import random
a = int(input())
b = []
l = []

try:
    for i in range(a): #создание базового массива
        l.append(random.randint(-3, 3))
    print(l)

    for i in range(1, len(l), 2): #создание нового массива
        b.append(l[i])
    print(b, len(b))
except:
    print('Error')

```

Протокол программы:

```
C:\Projects_1sem_Slusarev\venv\Scripts\python.exe C:\Projects_1sem_Slusarev\PZ_6\PZ_6_2.py
5
[3, 1, -1, 1, -3]
[1, 1] 2

Process finished with exit code 0
|
```

Задание 3

Постановка задачи.

1. Дано множество A из N точек ($N > 2$, точки заданы своими координатами x, y). Найти такую точку из данного множества, сумма расстояний от которой до остальных его точек минимальна, и саму эту сумму.

Расстояние R между точками с координатами (x_1, y_1) и (x_2, y_2) вычисляется по формуле:

$$R = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}.$$

Для хранения данных о каждом наборе точек следует использовать по два списка: первый список для хранения абсцисс, второй — для хранения ординат.

Тип алгоритма : Циклический

Текст программы:

```
# 3. Дано множество A из N точек (N > 2, точки заданы своими координатами
x, y).
# Найти такую точку из данного множества, сумма расстояний от которой до
остальных его точек минимальна,
# и саму эту сумму.
# Расстояние R между точками с координатами (x1, y1) и (x2, y2) вычисляется
по формуле:
# R = sqrt((x2 - x1)^2 + (y2 - y1)^2).
# Для хранения данных о каждом наборе точек следует использовать по два
списка:
# первый список для хранения абсцисс, второй — для хранения ординат.

import math

points = [(1, 2), (3, 4), (5, 6)]# точки

min_sum = float(100000000) # изначальное min sum
min_point = ()

try:
    # Нахождение точки с минимальной суммой расстояний
    for point in points:
        x, y = point
```

```

        total = 0
        for i in range(len(points)):
            total += math.sqrt((points[i][0] - x) ** 2 + (points[i][1] - y)
** 2)

        if total < min_sum:
            min_sum = total
            min_point = (x, y)

#Вывод
print("Точка с минимальной суммой расстояний :", min_point)
print("Минимальная сумма расстояний :", min_sum)
except:
    print('Error')

```

Протокол программы

```

C:\Projects_1sem_Slusarev\venv\Scripts\python.exe C:\Projects_1sem_Slusarev\PZ_6\PZ_6_3.py
Точка с минимальной суммой расстояний : (3, 4)
Минимальная сумма расстояний : 5.656854249492381

Process finished with exit code 0

```

Вывод:

в процессе выполнения задания выработал основные принципы составления программ, приобрести навыки сосоставления программ со списками в IDE PyCharm Community

Готовые решения выложены на Github.