

Практическая работа № 6.

Тема: составление программ со списками в IDE PyCharm Community

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 целочисленный список размера N. Проверить, чередуются ли в нем четные и нечетные числа. Если чередуются, то вывести 0, если нет, то вывести порядковый номер первого элемента, нарушающего закономерность.

Тип алгоритма: циклический

Текст программы:

```
# Дан целочисленный список размера N. Проверить, чередуются ли в нем четные и
# нечетные числа. Если чередуются, то вывести 0, если нет, то вывести порядковый
# номер первого элемента, нарушающего закономерность.

def Alternation(nums):
    for i in range(1, len(nums)):
        if nums[i] % 2 == nums[i-1] % 2:
            return i
    return 0

while True:
    try:
        string = input('Введите несколько чисел через пробел: ')
        string = string.split()
        nums = []
        for i in string:
            if i.isdigit():
                nums.append(int(i))
            else:
                raise ValueError

        result = Alternation(nums)
        print(result)
        break
    except ValueError:
        print('Ошибка! Введите числа.')
```

Протокол работы программы:

Введите несколько чисел через пробел: 1 2 3 4 5 6 7 8  
0

Process finished with exit code 0

Задание 2.

Постановка задачи.

Даны два списка A и B одинакового размера N. Сформировать новый список C того же размера, каждый элемент которого равен максимальному из элементов списков A и B.

Тип алгоритма: линейный

Текст программы:

```
# Даны два списка A и B одинакового размера N. Сформировать новый список C того
# же размера, каждый элемент которого равен максимальному из элементов списков
# A и B.
while True:
    try:
        C = []

        string = input('Введите числа через пробел: ')
        nums = string.split()
        A = []
        for i in nums:
            if i.isdigit():
                A.append(int(i))
            else:
                raise ValueError

        string = input('Введите числа через пробел: ')
        nums = string.split()
        B = []
        for i in nums:
            if i.isdigit():
                B.append(int(i))
            else:
                raise ValueError

        for i in range(len(A)):
            C.append((max([A[i], B[i]])))
        print(C)
        break
    except ValueError:
        print('Ошибка! Введите числа.')
```

Протокол работы программы:

Введите числа через пробел: 1 2 3 4 5 6 7 8

Введите числа через пробел: 2 1 5 3 4 6 8 4

[2, 2, 5, 4, 5, 6, 8, 8]

Process finished with exit code 0

Задание 3.

Постановка задачи.

Даны множества A и B, состоящие соответственно из N1 и N2 точек (точки заданы своими координатами x, y). Найти минимальное расстояние между точками этих множеств и сами точки, расположенные на этом расстоянии (вначале выводится точка из множества A, затем точка из множества B). Расстояние R между точками с координатами (x1, y1) и (x2, y2) 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:

```
# Даны множества A и B, состоящие соответственно из N1 и N2 точек (точки заданы
# своими координатами x, y). Найти минимальное расстояние между точками этих
```

```

# множеств и сами точки, расположенные на этом расстоянии (вначале выводится
# точка из множества A, затем точка из множества B).
# 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

def min_distance(A, B):
    min_dist = float('inf')
    min_point_A = None
    min_point_B = None

    for i in range(len(A)):
        for j in range(len(B)):
            dist = math.sqrt((B[j][0] - A[i][0])**2 + (B[j][1] - A[i][1])**2)
            if dist < min_dist:
                min_dist = dist
                min_point_A = A[i]
                min_point_B = B[j]

    return min_dist, min_point_A, min_point_B

A = [(1, 2), (3, 4), (5, 6)]
B = [(7, 8), (9, 10), (11, 12)]
min_dist, point_A, point_B = min_distance(A, B)
print(f"Минимальное расстояние: {min_dist}")
print(f"Точка из множества A: {point_A}")
print(f"Точка из множества B: {point_B}")

```

Протокол работы программы:

Минимальное расстояние: 2.8284271247461903

Точка из множества A: (5, 6)

Точка из множества B: (7, 8)

Process finished with exit code 0

Вывод: в процессе выполнения практического занятия выработал(а) навыки составления программ со списками в IDE PyCharm Community. Были использованы языковые конструкции isdigit, append.

Готовые программные коды выложены на GitHub.