Студент группы ИС-27 Прудников А.Г.

Практическое занятие №6

Тема: Составление программ со списками в IDE PyCharm Community.

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

Постановка задачи.

1. Дан список размера N и целые числа К и L (1 < К L S N). Найти средне арифметическое всех элементов списка, кромс элементов с номерами от К до включительно.

2. Даны списки А и В одинакового размера N. Поменять местами их содержимое и вывести вначале элементы преобразованного списка А, а затем элементы преобразовавшего списка В.

3. Дано множество А из N точек на плоскости и точка В (точки заданы своими координатами х, у). Найти точку из множества А, наиболее близкую к точке В.

Расстояние R между точками с координатами (XI, у) и (м, у) вычисляется по формуле:

R = √ (х 2 - х 1):2 + (у 2 - у 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: Линейный

Текст программы:

```
1) # Дан список размера N и целые числа K и L (1 < K ⩽ L ⩽ N).
Найти среднее арифметическое всех элементов
массива, кроме

# элементов с номерами от K до L включительно.

N = [1, 2, 3, 4, 5]

k = 0

m = 0

r = [x for i, x in enumerate(N, 1) if i < k or i > m]

print(r)

print(sum(r) / len(r))
```

2) # Даны списки А и В одинакового размера N. Поменять местами их содержимое и

вывести вначале элементы преобразованного списка A, а затем

элементы

преобразованного списка B.

a = list(range(1, 11))

b = list(range(10, 0, -1))

a, b = b, a

print(b, a, a, b, sep='\n')

3) # Дано множество A из N точек на плоскости и точка B (точки заданы своими

координатами x, y). Найти точку из множества A, наиболее близкую к точке B.

Расстожше R между точками с координатами (XI, y) и (м, y) вычисляется по

формуле:

R = √ (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.

import math

def find_closest_point(A):

    closest_point = [0, 0] # Изначально считаем точку с нулевыми координатами ближайшей

    closest_distance = math.inf # Изначально считаем бесконечную дистанцию

    for i in range(len(A[0])):

        x = A[0][i]

        y = A[1][i]

        # Проверяем, что точка лежит в первой или третьей четверти

        if x >= 0 and y >= 0 or x <= 0 and y <= 0:

            distance = math.sqrt(x ** 2 + y ** 2) # Вычисляем расстояние до начала координат

            if distance < closest_distance:

                closest_distance = distance

```
            closest_point = [x, y]

    return closest_point


# Пример использования функции
A = [[1, 2, 3, 4, 5], [6, 7, 8, 9, 10]]
closest_point = find_closest_point(A)
print("Ближайшая точка:", closest_point)
```

Протокол работы программы:

1)

```
[1, 2, 3, 4, 5]
3.0

Process finished with exit code 0
```

2)

```
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
[10, 9, 8, 7, 6, 5, 4, 3, 2, 1]
[10, 9, 8, 7, 6, 5, 4, 3, 2, 1]
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10]

Process finished with exit code 0
```

3)

```
Ближайшая точка: [1, 6]

Process finished with exit code 0
```

Вывод: Научился работать с листами в процессе выполнения практической.