

Отчет по практической работе

Практическое занятие № 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.

ЗАДАЧА 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

n = int(input('Введите количество элементов списка: '))

l = ['n'] * n # создание списка

i = 0 # счетчик

while i != n: # заполнение списка

    l[i] = int(input('Введите число: '))

    i += 1

i = 0 # обнуление счетчика
k = 0 # индикатор

while i != n - 1: # проверка списка

    if l[i] % 2 == l[i + 1] % 2:

        print (i + 1)

        k += 1

        break

    else:

        i += 1

        continue
```

```
if k == 0: # проверка индикатора

    print(0)
```

Протокол работы программы:

Введите количество элементов списка: 4

Введите число: 1

Введите число: 3

Введите число: 7

Введите число: 6

1

Process finished with exit code 0

ЗАДАЧА 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: циклический

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

def max(a, b): # функция нахождения максимального числа из двух
    if a > b:
        return a
    elif b > a:
        return b
    else:
        return a

n = int(input('Введите количество элементов списков A и B: '))
a, b, c = ['n'] * n, ['n'] * n, ['n'] * n # создание списка

i = 0 # счетчик нет
print('Заполнение списка A')
while i != n: # заполнение списка
    a[i] = int(input('Введите число: '))
    i += 1

i = 0 # обнуление счетчика
print('Заполнение списка B')
while i != n: # заполнение списка
    b[i] = int(input('Введите число: '))
    i += 1
```

```

i = 0 # обнуление счетчика
while i != n: # заполнение списка
    c[i] = max(a[i], b[i])
    i += 1

print('Список C:', c)

```

Протокол работы программы:

Введите количество элементов списков A и B: 4

Заполнение списка A

Введите число: 7

Введите число: 1

Введите число: 6

Введите число: 4

Заполнение списка B

Введите число: 1

Введите число: 3

Введите число: 2

Введите число: 5

Список C: [7, 3, 6, 5]

Process finished with exit code 0

ЗАДАЧА 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: $R = \sqrt{(x2 - x1)^2 + (y2 - y1)^2}$. Для хранения данных о каждом наборе точек следует использовать по два списка: первый список для хранения абсцисс, второй — для хранения ординат.

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

```

```

from math import sqrt

n1 = int(input('Введите количество точек множества A: '))
n2 = int(input('Введите количество точек множества B: '))
ax, bx = ['n'] * n1, ['n'] * n2 # создание списков абсцисс
ay, by = ['n'] * n1, ['n'] * n2 # создание списков ординат

i = 0 # счетчик
print('\nЗаполнение списка абсцисс множества A')
while i != n1: # заполнение списка
    ax[i] = int(input('Введите число: '))
    i += 1

i = 0 # обнуление счетчика
print('\nЗаполнение списка ординат множества A')
while i != n1: # заполнение списка
    ay[i] = int(input('Введите число: '))
    i += 1

i = 0 # обнуление счетчика
print('\nЗаполнение списка абсцисс множества B')
while i != n2: # заполнение списка
    bx[i] = int(input('Введите число: '))
    i += 1

i = 0 # обнуление счетчика
print('\nЗаполнение списка ординат множества B')
while i != n2: # заполнение списка
    by[i] = int(input('Введите число: '))
    i += 1

i = 0
k = 0
c = 0
r = sqrt((int(bx[k]) - int(ax[i])) ** 2 + (int(by[k]) - int(ay[i])) ** 2)
ia = 0
ib = 0
while i != n1:
    while k != n2:
        r = sqrt((int(bx[k]) - int(ax[i])) ** 2 + (int(by[k]) - int(ay[i])) **
2)
        if r < c:
            c = r
            axz, ayz, bxz, byz = ax[i], ay[i], bx[k], by[k]
            ia = i
            ib = k
            k += 1
        i += 1

i = 0
k = 0
while i != n2:
    while k != n1:
        r = sqrt((int(bx[i]) - int(ax[k])) ** 2 + (int(by[i]) - int(ay[k])) **
2)
        if r > c:
            c = r
            axz, ayz, bxz, byz = ax[k], ay[k], bx[i], by[i]
            ia = k
            ib = i
            k += 1
        i += 1

print('\nНаименьшее расстояние между точками =', c)

```

```
print('\nЭти точки: A[', ia, ']( ', axz, '; ', ayz, ') B[',  
      ib, ']( ', bxz, '; ', byz, ')', sep='')
```

Протокол работы программы:

Введите количество точек множества A: 3

Введите количество точек множества B: 2

Заполнение списка абсцисс множества A

Введите число: 1

Введите число: 4

Введите число: 2

Заполнение списка ординат множества A

Введите число: 6

Введите число: 5

Введите число: 3

Заполнение списка абсцисс множества B

Введите число: 2

Введите число: 11

Заполнение списка ординат множества B

Введите число: 3

Введите число: 8

Наименьшее расстояние между точками = 3.1622776601683795

Эти точки: A[0](1;6) B[0](2;3)

Process finished with exit code 0

Вывод: В процессе выполнения практической работы я закрепил усвоенные знания, понятия, алгоритмы, основные принципы составления программ, приобрел навыки составления программ со списками в IDE PyCharm Community.

Были использованы языковые конструкции def, while, if, elif, else

Выполнены разработка кода, отладка, тестирование, оптимизация программного кода.

Готовые программные коды выложены на GitHub.