

Отчёт по практической работе №6

Вариант 26

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.

Постановка задачи №1: 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 первого элемента, нарушающего закономерность.

```
# 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 первого
элемента, нарушающего закономерность.

N = int(input("N:"))      # Ввод длины списка
spisok = []
for i in range(N):        # функция позволяющая заполнить список вручную
    spisok.append(int(input("Порядковый номер " + str(i + 1) + " = "))) # i +
1, потому что порядок в списке начинается                               # с 0,
а порядковые номера с 1.
k=0
for i in range(len(spisok) - 1): # функция вычисляющая порядок списка
    if spisok[i]<0 and spisok[i+1]>=0:
        continue
    elif spisok[i]>=0 and spisok[i+1]<0:
        continue
    else:
        k=i+2
        break
print(k)
```

Протокол выполнения программы №1:

N:6

Порядковый номер 1 = 1

Порядковый номер 2 = -12321

Порядковый номер 3 = 123123123

Порядковый номер 4 = -412

Порядковый номер 5 = 132

Порядковый номер 6 = -321

0

Process finished with exit code 0

Протокол выполнения программы №2:

N:6

Порядковый номер 1 = 2

Порядковый номер 2 = -432

Порядковый номер 3 = 234

Порядковый номер 4 = 243

Порядковый номер 5 = -432

Порядковый номер 6 = 234

4

Process finished with exit code 0

Постановка задачи №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.

```
# 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.

N = int(input("N:"))      # Ввод длины списка
while N > 14:
    N -=1

A = []
for i in range(N):        # функция позволяющая заполнить список вручную
    A.append(int(i) + 1)

print("Данный список A = ",A)

B = A[::2]                # копирование списка с шагом 2

print("Новый список с нечётными элементами первого списка = ",B)
print("Размер полученного списка = ",len(B))
```

Протокол выполнения программы №1:

N:123

Данный список A = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14]

Новый список с нечётными элементами первого списка = [1, 3, 5, 7, 9, 11, 13]

Размер полученного списка = 7

Process finished with exit code 0

Протокол выполнения программы №2:

N:12

Данный список A = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12]

Новый список с нечётными элементами первого списка = [1, 3, 5, 7, 9, 11]

Размер полученного списка = 6

Process finished with exit code 0

Постановка задачи №2: 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:

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

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

```

# 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.

from math import *      # Импорт функций из библиотеки math

N = int(input("Введите кол-во точек: "))
X = []
Y = []
for i in range(N):      # Функция позволяющая пользователю ручной ввод
    X.append(int(input("Введите координату X" + str(i + 1) + ":")))
    Y.append(int(input("Введите координату Y" + str(i + 1) + ":")))

sum = []                # Список сумм

for h in range(N):      # Функция нахождения сумм расстояний
    S = 0
    for i in range(N - 1):
        R = (sqrt(pow((X[0] - X[i+1]),2) + pow((Y[0]-Y[i+1]),2)))
        S += R

    X.append(X[0])      # Прокрутка списка координат
    Y.append(Y[0])
    X = X[1:]
    Y = Y[1:]
    sum.append(round(S , 2)) # Ввод в список сумм с округлением до сотых

print("Минимальная сумма расстояний: ",min(sum))
print("Координаты точки: (", X[sum.index(min(sum))],";", Y[sum.index(min(sum))],")")

```

Протокол выполнения программы №1:

Введите кол-во точек: 4
Введите координату X1:10
Введите координату Y1:12
Введите координату X2:32
Введите координату Y2:12
Введите координату X3:32
Введите координату Y3:4
Введите координату X4:3
Введите координату Y4:23
Минимальная сумма расстояний: 58.45
Координаты точки: (10 ; 12)
Process finished with exit code 0

Протокол выполнения программы №2:

Введите кол-во точек: 6
Введите координату X1:123
Введите координату Y1:321
Введите координату X2:-123
Введите координату Y2:321
Введите координату X3:-321
Введите координату Y3:12
Введите координату X4:-32
Введите координату Y4:12
Введите координату X5:-234
Введите координату Y5:-74
Введите координату X6:12
Введите координату Y6:123
Минимальная сумма расстояний: 1252.21
Координаты точки: (12 ; 123)
Process finished with exit code 0

Вывод: закрепил знания, основные принципы составления программ, приобрел навыки составление программ со списками в IDE PyCharm Community.

Использованные конструкции: for, import, while, if.