

Практическое задание №6

Тема: составление программ со списками в IDE PyCharm Community.

Цель: закрепить усвоенные знания, понятия, алгоритмы, основные принципы составления программ, приобрести навыки составления программ св IDE PyCharm Community.

Постановка задачи №1:

Дан список A размера N (N — четное число). Вывести его элементы с четными номерами в порядке возрастания номеров: A₂, A₄, A₆, ..., A_N. Условный оператор не использовать.

Тип алгоритма: циклический

Текст программы:

```
import random

def spisok():
    try:
        N = random.randint(5, 10) * 2 # Генерируем четное число N
        A = [random.randint(1, 100) for _ in range(N)] # Генерируем список
        случайных чисел

        result = sorted(A[1::2]) # Получаем элементы с четными индексами и
        сортируем их по возрастанию

        print("Число N:", N)
        print("Сгенерированный список A:", A) # Выводим сгенерированный
        список A
        print("Элементы с четными индексами по возрастанию:", result) #
        Выводим результат

    except ValueError as e:
        print("Ошибка:", e)
        spisok()

spisok()
```

Протокол работы программы:

Число N: 14

Сгенерированный список A: [81, 67, 86, 59, 49, 28, 27, 77, 61, 93, 92, 54, 31, 40]

Элементы с четными индексами по возрастанию: [28, 40, 54, 59, 67, 77, 93]

Process finished with exit code 0

Постановка задачи №2:

Дано число R и список A размера N. Найти элемент списка, который наиболее близок к число R (то есть такой элемент A_k , для которого величина $|A_k - R|$ является минимальной)

Тип алгоритма: циклический

Текст программы:

```
import random

def closest_el(R, A):
    closest = A[0] # Предположим, что первый элемент наиболее близкий
    min_dif = abs(closest - R) # Разница между первым элементом и R

    for element in A:
        dif = abs(element - R) # Находим разницу между текущим элементом и R
        if dif < min_dif: # Если разница меньше наименьшей до этого
            min_dif = dif # Обновляем наименьшую разницу
            closest = element # Текущий элемент становится "наиболее близким"

    return closest # Возвращаем наиболее близкий элемент

# Получаем число R
R = float(input("Введите число R: "))

# Генерируем список случайных чисел
N = 10 # Длина списка
A = [random.randint(1, 100) for _ in range(N)] # Генерируем список случайных чисел

result = closest_el(R, A)
print(f"Наиболее близкий элемент к {R} из списка {A}: {result}")
```

Протокол работы программы:

Введите число R: 15

Наиболее близкий элемент к 15.0 из
списка [33, 32, 57, 62, 32, 18, 8, 50, 45, 1]: 18

Process finished with exit code 0

Постановка задачи №3:

Дан список размера N. Осуществить сдвиг элементов списка влево на одну позицию (при этом A_n перейдет в A_{n-1} , A_{n-1} – в A_{n-2} , ..., A_2 – в A_1 , а исходное значение первого элемента будет потеряно). Последний элемент полученного списка положить равным 0

Вывод: в процессе выполнения практического занятия закрепил усвоенные знания, понятия, алгоритмы, основные принципы составления программ, приобрести навыки составления программ с функциями в IDE PyCharm Community. Выполнены разработка кода, отладка, тестирование, оптимизация программного кода. Готовые программные коды выложены на GitHub.