**Студент группы ИС-28 Алияров Р.С.**

**Практическое занятие №6**

**Тема:** **составление программ со списками в IDE PyCharm Community.**

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

**Постановка задачи 6.1: Сформировать и вывести целочисленный список размера 10, содержащий 10 первых положительных нечетных чисел: 1,3,5, ... .**

**Текст программы:**

# Создаем список для хранения нечетных чисел

odd\_numbers = []

# Заполняем список первыми 10 положительными нечетными числами

for i in range(1, 20, 2): # Начинаем с 1, до 20 с шагом 2

odd\_numbers.append(i)

# Выводим список

print(odd\_numbers)

**Протокол работы задачи:**

**[1, 3, 5, 7, 9, 11, 13, 15, 17, 19]**

**Постановка задачи 6.2: Дан список размера N. Найти номера тех элементов списка, которые больше своего правого соседа, и количество таких элементов. Найденные номера выводить в порядке их возрастания.**

**Текст программы:**

def find\_greater\_than\_right\_neighbor(lst):

# Список для хранения индексов элементов, которые больше своего правого соседа

indices = []

# Проходим по списку, исключая последний элемент

for i in range(len(lst) - 1):

if lst[i] > lst[i + 1]:

indices.append(i)

# Выводим количество таких элементов и их индексы

print("Количество элементов, которые больше своего правого соседа:", len(indices))

print("Индексы таких элементов:", indices)

#Переменные

N = int(input("Введите размер списка N: "))

lst = []

# Заполнение списка

for i in range(N):

element = int(input(f"Введите элемент {i + 1}: "))

lst.append(element)

find\_greater\_than\_right\_neighbor(lst)

**Протокол работы программы**

**Введите размер списка N: 5**

**Введите элемент 1: 2**

**Введите элемент 2: 6**

**Введите элемент 3: 3**

**Введите элемент 4: 4**

**Введите элемент 5: 5**

**Количество элементов, которые больше своего правого соседа: 1**

**Индексы таких элементов: [1]**

**Постановка задачи 6.3: Дан список A размера N и целые числа K и L (1 < K < L < N). Переставить в обратном порядке элементы списка, расположенные между элементами AK и AL, не включая эти элементы**

**Текст программы:**

def reverse\_sublist(A, K, L):

# Проверяем, что K и L находятся в допустимых пределах

if K < 1 or L >= len(A) or K >= L:

print("Ошибка: K и L должны удовлетворять условиям 1 < K < L < N.")

return A

# Переворачиваем подсписок между K и L

A[K:L] = A[K:L][::-1]

return A

#Переменные

N = int(input("Введите размер списка N: "))

A = []

# Заполнение списка

for i in range(N):

element = int(input(f"Введите элемент {i + 1}: "))

A.append(element)

K = int(input("Введите K (1 < K < N): ")) - 1 # Преобразуем в 0-индексацию

L = int(input("Введите L (K < L < N): ")) - 1 # Преобразуем в 0-индексацию

# Переставляем элементы и выводим результат

result = reverse\_sublist(A, K, L)

print("Результат:", result)

**Протокол работы программы:**

**Введите размер списка N: 4**

**Введите элемент 1: 6**

**Введите элемент 2: 3**

**Введите элемент 3: 4**

**Введите элемент 4: 2**

**Введите K (1 < K < N): 9**

**Введите L (K < L < N): 8**

**Ошибка: K и L должны удовлетворять условиям 1 < K < L < N.**

**Результат: [6, 3, 4, 2]**

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