Министерство образования Республики Беларусь
Учреждение образования
«Брестский государственный технический университет»
Кафедра ИИТ

Лабораторная работа №1

По дисциплине: “СПП”

Выполнил:
Студент 3 курса
Группы ПО-12
Боричевсвкий Д. О.
Проверила:
Кулик А. Д.

Брест 2026

Цель работы: Закрепление навыков объектно-ориентированного программирования на языке Python.

Задание 1

2) Вывод максимального и минимального значения, а также суммы и произведения элементов последовательности.

Решение:

```
def process_sequence():
    # Ввод последовательности
    n = int(input("Введите количество элементов N: "))
    sequence = []

    print(f"Введите {n} целых чисел:")
    for i in range(n):
        num = int(input(f"Элемент {i + 1}: "))
        sequence.append(num)

    # Вычисления
    max_val = max(sequence)
    min_val = min(sequence)
    sum_val = sum(sequence)

    # Произведение (с защитой от переполнения для больших чисел)
    prod_val = 1
    for num in sequence:
        prod_val *= num

    # Вывод результатов
    print(f"\nРезультаты обработки последовательности {sequence}:")
    print(f"Максимальное значение: {max_val}")
    print(f"Минимальное значение: {min_val}")
    print(f"Сумма элементов: {sum_val}")
    print(f"Произведение элементов: {prod_val}")


# Запуск
process_sequence()
```

Задание 2

2) Даны два целочисленных списка nums1 и nums2, отсортированных в неубывающем порядке, и два целых числа m и n, представляющих количество элементов в nums1 и nums2 соответственно. Объедините nums1 и nums2 в один список, отсортированный в неубывающем порядке. Окончательный отсортированный список не должен возвращаться функцией, а вместо этого должен храниться внутри списка nums1. Чтобы учесть это, nums1 имеет длину m + n, где первые m элементов обозначают элементы, которые должны быть объединены, а последние n элементов устанавливаются в 0 и должны

игнорироваться. nums2 имеет длину n.
Input: nums1 = [1,2,3,0,0,0], m = 3, nums2 = [2,5,6], n = 3
Output: [1,2,2,3,5,6]

Решение:

```
def merge(nums1, m, nums2, n):

    # Указатели на текущие позиции (с конца)
    i = m - 1  # последний значимый элемент в nums1
    j = n - 1  # последний элемент в nums2
    k = m + n - 1  # позиция для вставки в nums1

    # Пока есть элементы в обоих списках
    while i >= 0 and j >= 0:
        if nums1[i] > nums2[j]:
            nums1[k] = nums1[i]
            i -= 1
        else:
            nums1[k] = nums2[j]
            j -= 1
        k -= 1

    # Если остались элементы в nums2 (в nums1 они уже на месте)
    while j >= 0:
        nums1[k] = nums2[j]
        j -= 1
        k -= 1

    return nums1


def input_sorted_list(name, count, allow_zeros=False):
    """Ввод отсортированного списка с проверкой"""
    print(f"\nВвод списка {name} ({count} элементов):")
    result = []
    for i in range(count):
        while True:
            try:
                num = int(input(f"  Элемент {i + 1}: "))
                # Проверка на неубывающий порядок
                if i > 0 and num < result[-1]:
                    print("  Ошибка: список должен быть отсортирован по
неубыванию!")
                    continue
                result.append(num)
                break
            except ValueError:
                print("  Ошибка: введите целое число!")

    # Если нужно добавить нули (для nums1)
    if allow_zeros:
        result.extend([0] * allow_zeros)
        print(f"  (добавлено {allow_zeros} нулей в конец для слияния)")

    return result


def main():
    print("=" * 50)
    print("СЛИЯНИЕ ДВУХ ОТСОРТИРОВАННЫХ СПИСКОВ")
    print("=" * 50)
```

```
    # Ввод параметров
    print("\n--- Ввод параметров ---")

    # Ввод m и n
    while True:
        try:
            m = int(input("Введите m (количество элементов в nums1): "))
            n = int(input("Введите n (количество элементов в nums2): "))
            if m < 0 or n < 0:
                print("Числа должны быть неотрицательными!")
                continue
            break
        except ValueError:
            print("Введите целые числа!")

    # Ввод nums1 (только m значимых элементов)
    nums1 = input_sorted_list("nums1", m, allow_zeros=n)

    # Ввод nums2 (n элементов)
    nums2 = input_sorted_list("nums2", n)

    # Вывод исходных данных
    print("\n" + "=" * 50)
    print("ИСХОДНЫЕ ДАННЫЕ:")
    print(f"  nums1 = {nums1}")
    print(f"  m = {m}")
    print(f"  nums2 = {nums2}")
    print(f"  n = {n}")

    # Выполнение слияния
    merge(nums1, m, nums2, n)

    # Вывод результата
    print("\n" + "=" * 50)
    print("РЕЗУЛЬТАТ:")
    print(f"  nums1 = {nums1}")
    print("=" * 50)


# Запуск программы
if __name__ == "__main__":
    main()
```

Вывод: В результате выполнения лабораторной работы были закреплены знания объектно-ориентированного программирования Python при решении задач.