

МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ
УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ
«БРЕСТСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»
ФАКУЛЬТЕТ ЭЛЕКТРОННО-ИНФОРМАЦИОННЫХ СИСТЕМ
Кафедра интеллектуальных информационных технологий

Отчёт по лабораторной работе №1

Специальность ПО11

Выполнил
Н. А. Антонюк
студент группы ПО11

Проверил
А. А. Крощенко
ст. преп. кафедры ИИТ,
22.02.2025 г.

Брест 2025

Цель работы: закрепить базовые знания языка программирования Python при решении практических задач

Задание 1. Вывод максимального и минимального значения, а также суммы и произведения элементов последовательности.

Выполнение:

Код программы:

```
def process_sequence(sequence):
    if not sequence:
        return "The sequence is empty"

    max_value = max(sequence)
    min_value = min(sequence)
    sum_value = sum(sequence)

    product_value = 1
    for num in sequence:
        product_value *= num

    return max_value, min_value, sum_value, product_value

# Example usage
N = int(input("Enter the number of elements in the sequence: "))
sequence = [int(input(f"Enter element {i+1}: ")) for i in range(N)]

max_val, min_val, sum_val, product_val = process_sequence(sequence)

print(f"Maximum value: {max_val}")
print(f"Minimum value: {min_val}")
print(f"Sum of elements: {sum_val}")
print(f"Product of elements: {product_val}")
```

Спецификация ввода:

<Количество элементов последовательности> <1-й элемент массива> ... <n-й элемент массива>

Пример:

5 5 7 9 1 3

Спецификация вывода:

Maximum value: {< Значение максимального элемента массива >}
Minimum value: {< Значение минимального элемента массива >}
Sum of elements: {< Значение суммы элементов массива >}
Product of elements: {< Значение произведения элементов массива >}

Пример:

Maximum value: 9
Minimum value: 1
Sum of elements: 25
Product of elements: 945

Рисунки с результатами работы программы:

```
PS C:\Users\Nikita> & C:/Users/Nikita/AppData/Local/Programs/Python/Python311-32/python.exe c:/Users/Nikita/Desktop/Untitled-1.py
Enter the number of elements in the sequence: 5
Enter element 1: 5
Enter element 2: 7
Enter element 3: 9
Enter element 4: 1
Enter element 5: 3
Maximum value: 9
Minimum value: 1
Sum of elements: 25
Product of elements: 945
PS C:\Users\Nikita> 
```

```
PS C:\Users\Nikita> & C:/Users/Nikita/AppData/Local/Programs/Python/Python311-32/python.exe c:/Users/Nikita/Desktop/Untitled-1.py
Enter the number of elements in the sequence: 7
Enter element 1: 1
Enter element 2: 2
Enter element 3: 3
Enter element 4: 4
Enter element 5: 5
Enter element 6: 6
Enter element 7: 7
Maximum value: 7
Minimum value: 1
Sum of elements: 28
Product of elements: 5040
```

Задание 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]`

Выполнение:

Код программы:

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

```
    p1 = m - 1
```

```
    p2 = n - 1
```

```
    p = m + n - 1
```

```
    while p1 >= 0 and p2 >= 0:
```

```
        if nums1[p1] > nums2[p2]:
```

```
            nums1[p] = nums1[p1]
```

```
            p1 -= 1
```

```

else:
    nums1[p] = nums2[p2]
    p2 -= 1
    p -= 1

while p2 >= 0:
    nums1[p] = nums2[p2]
    p2 -= 1
    p -= 1

m = int(input("Введите количество элементов в nums1 (m): "))
n = int(input("Введите количество элементов в nums2 (n): "))

print("Введите элементы списка nums1 (отсортированного в неубывающем порядке):")
nums1 = []
for i in range(m):
    nums1.append(int(input(f"nums1[{i}]: ")))

nums1.extend([0] * n)

print("Введите элементы списка nums2 (отсортированного в неубывающем порядке):")
nums2 = []
for i in range(n):
    nums2.append(int(input(f"nums2[{i}]: ")))

merge(nums1, m, nums2, n)

print("Объединенный и отсортированный список nums1:", nums1)

```

Спецификация ввода:

<Количество элементов в nums1>

<Количество элементов в nums2> <1 элемент 1ого массива>... <n-й элемент 1ого массива>

<1 элемент 2ого массива>... <n-й элемент 2ого массива>

Пример:

3 3 1 2 3 4 5 6

Спецификация вывода:

Введите количество элементов в nums1 (m): {< Количество элементов в nums1 >}

Введите количество элементов в nums2 (n): {< Количество элементов в nums2 >}

Введите элементы списка nums1 (отсортированного в неубывающем порядке):

nums1[0]: {< 1 элемент m-ого массива >}

nums1[1]: {< 1 элемент m-ого массива >}

nums1[2]: {< 1 элемент m-ого массива >}

Введите элементы списка nums2 (отсортированного в неубывающем порядке):

nums2[0]: {< 1 элемент n-ого массива >}

nums2[1]: {< 2 элемент n-ого массива >}

nums2[2]: {< 3 элемент n-ого массива >}

Объединенный и отсортированный список nums1: {< Значение произведения элементов массива >}

Пример:

Введите количество элементов в nums1 (m): 3

Введите количество элементов в nums2 (n): 3

Введите элементы списка nums1 (отсортированного в неубывающем порядке):

nums1[0]: 1

nums1[1]: 2

nums1[2]: 3

Введите элементы списка nums2 (отсортированного в неубывающем порядке):

nums2[0]: 4

nums2[1]: 6

nums2[2]: 7

Объединенный и отсортированный список nums1: [1, 2, 3, 4, 6, 7]

Рисунки с результатами работы программы:

```
Введите количество элементов в nums1 (m): 3
Введите количество элементов в nums2 (n): 3
Введите элементы списка nums1 (отсортированного в неубывающем порядке):
nums1[0]: 1
nums1[1]: 2
nums1[2]: 3
Введите элементы списка nums2 (отсортированного в неубывающем порядке):
nums2[0]: 4
nums2[1]: 6
nums2[2]: 7
Объединенный и отсортированный список nums1: [1, 2, 3, 4, 6, 7]
PS C:\Users\Nikita>
```