

Министерство науки и высшего образования Российской Федерации
Федеральное государственное автономное образовательное учреждение
высшего образования
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт цифрового развития
Кафедра инфокоммуникаций

ОТЧЕТ
ПО РАБОТЕ №2.4
дисциплины «Основы кроссплатформенного программирования»
Вариант 3

Выполнил:
Баканов Артем Вадимович
1 курс, группа ИТС-6-0-22-1,
11.03.02 «Инфокоммуникационные
технологии и системы связи»,
направленность (профиль)
«Инфокоммуникационные системы и
сети», очная форма обучения

(подпись)

Руководитель практики:
Воронкин Р.А., канд. тех. наук, доцент,
доцент кафедры инфокоммуникаций

(подпись)

Отчет защищен с оценкой _____ Дата защиты _____

Ставрополь, 2023 г.

Тема: работа со строками в языке Python.

Цель работы: приобретение навыков по работе со строками при написании программ с помощью языка программирования Python версии 3.x.

Ссылка: <https://github.com/temacteklyannayapuwka/cross-platform-programming-v.2.4>

Порядок выполнения работы:

1. Создал общедоступный репозиторий на GitHub, в котором будет использована лицензия MIT.

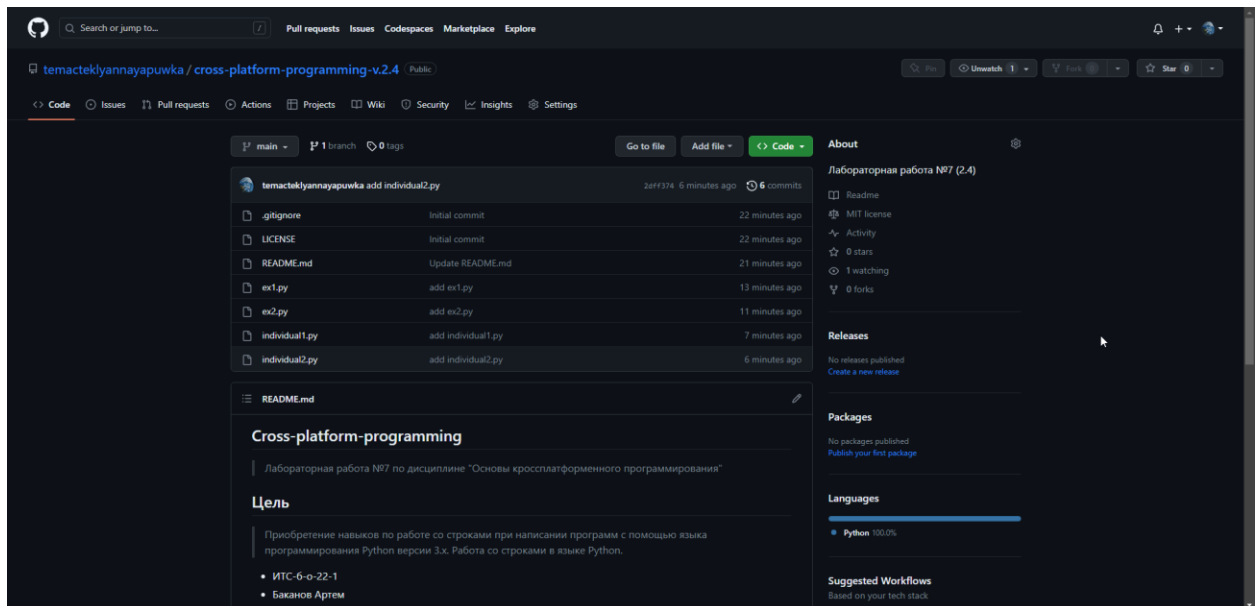


Рис. 1. Новый репозиторий.

2. В ходе данной лабораторной работы работал с моделью ветвления gitflow.

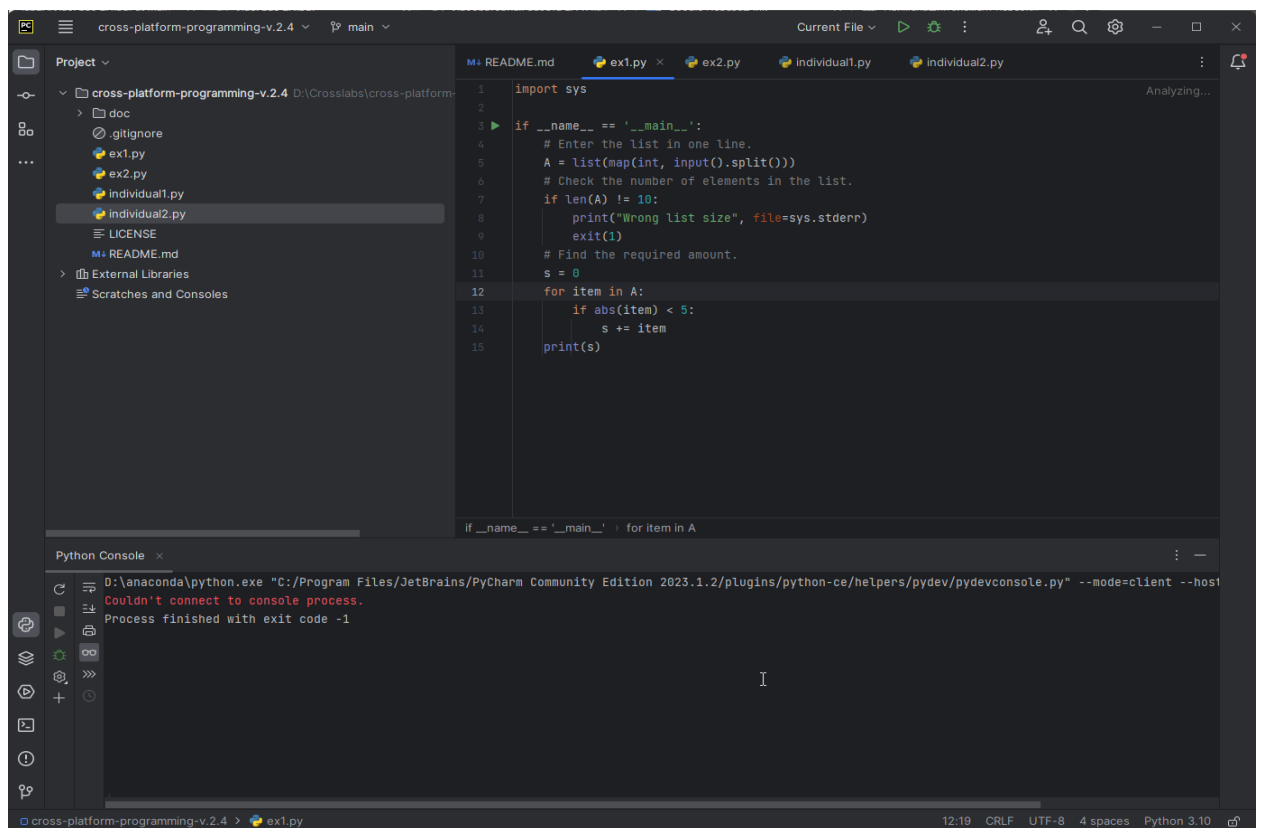
```
PS D:\> cd D:\Crosslabs
PS D:\Crosslabs> git clone https://github.com/temacteklyannayapuwka/cross-platform-programming-v.2.4.git
Cloning into 'cross-platform-programming-v.2.4'...
remote: Enumerating objects: 8, done.
remote: Counting objects: 100% (8/8), done.
remote: Compressing objects: 100% (8/8), done.
remote: Total 8 (delta 1), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (8/8), done.
Resolving deltas: 100% (1/1), done.
PS D:\Crosslabs> cd cross-platform-programming-v.2.4
PS D:\Crosslabs\cross-platform-programming-v.2.4> git flow init

Which branch should be used for bringing forth production releases?
- main
Branch name for production releases: [main]
Branch name for "next release" development: [develop]

How to name your supporting branch prefixes?
Feature branches? [feature/]
Bugfix branches? [bugfix/]
Release branches? [release/]
Hotfix branches? [hotfix/]
Support branches? [support/]
Version tag prefix? []
Hooks and filters directory? [D:/Crosslabs/cross-platform-programming-v.2.4/.git/hooks]
PS D:\Crosslabs\cross-platform-programming-v.2.4> git branch
* develop
main
```

Рис. 2. Клонирование и модель ветвления git-flow.

3. Пример №1. Условие примера: Ввести список А из 10 элементов, найти сумму элементов, меньших по модулю 5, и вывести ее на экран.



The screenshot shows the PyCharm IDE interface. The main editor displays a Python script named `ex1.py` with the following code:

```
1 import sys
2
3 if __name__ == '__main__':
4     # Enter the list in one line.
5     A = list(map(int, input().split()))
6     # Check the number of elements in the list.
7     if len(A) != 10:
8         print("Wrong list size", file=sys.stderr)
9         exit(1)
10    # Find the required amount.
11    s = 0
12    for item in A:
13        if abs(item) < 5:
14            s += item
15    print(s)
```

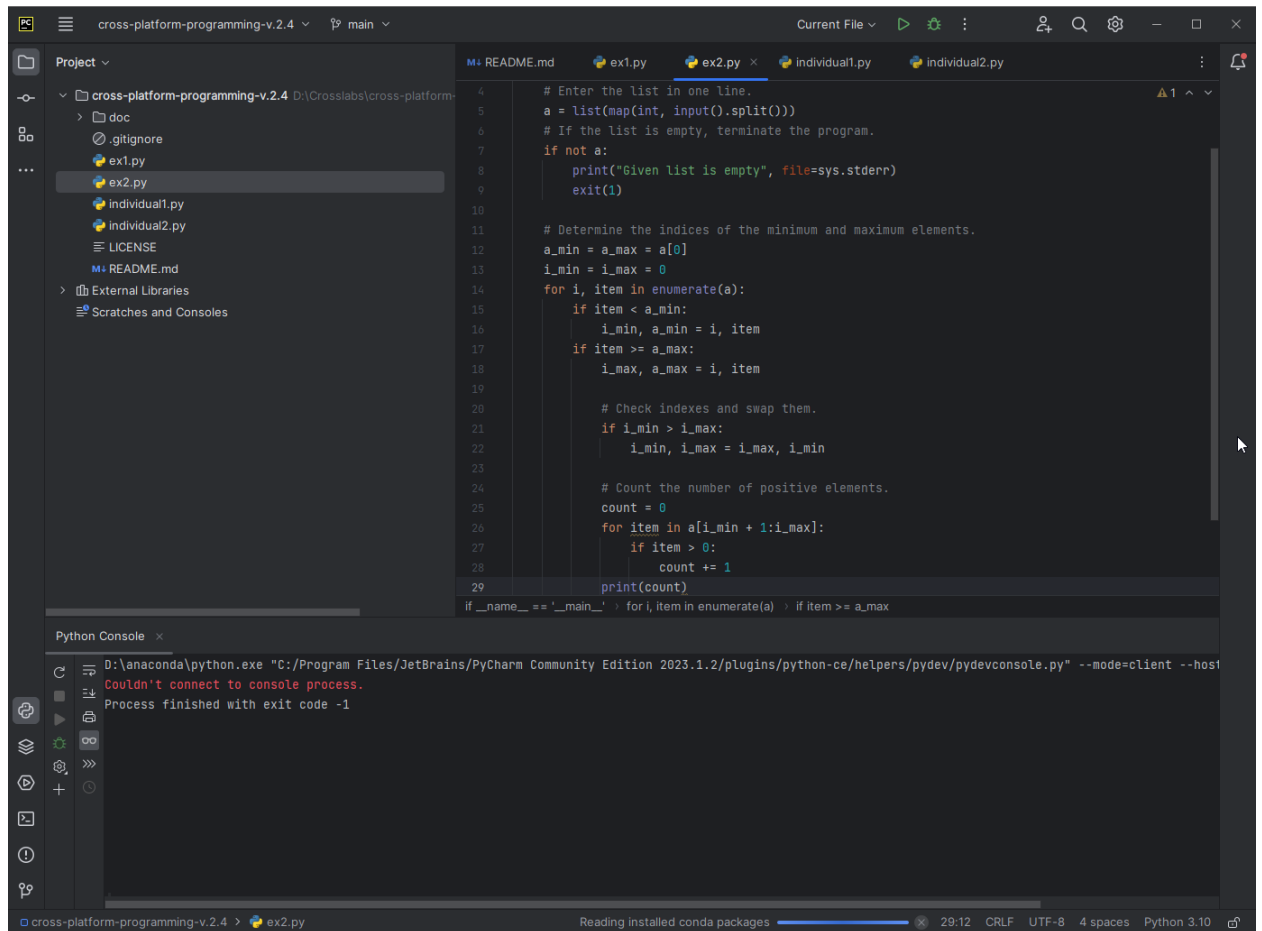
The left sidebar shows the project structure for `cross-platform-programming-v.2.4`, including files like `ex1.py`, `ex2.py`, `individual1.py`, and `individual2.py`.

The bottom panel shows the Python Console output:

```
D:\anaconda\python.exe "C:/Program Files/JetBrains/PyCharm Community Edition 2023.1.2/plugins/python-ce/helpers/pydev/pydevconsole.py" --mode=client --host=
Couldn't connect to console process.
Process finished with exit code -1
```

Рис. 3. Программа и ее результат.

4. Пример №2. Условие примера: написать программу, которая для целочисленного списка определяет, сколько положительных элементов располагается между его максимальным и минимальным элементами.



The screenshot shows the PyCharm IDE interface. The top toolbar includes icons for file operations, running, and debugging. The left sidebar displays the project structure for 'cross-platform-programming-v.2.4', with files like 'ex1.py', 'ex2.py', 'individual1.py', and 'individual2.py'. The main editor window shows the code for 'ex2.py'.

```
4 # Enter the list in one line.  
5 a = list(map(int, input().split()))  
6 # If the list is empty, terminate the program.  
7 if not a:  
8     print("Given list is empty", file=sys.stderr)  
9     exit(1)  
10  
11 # Determine the indices of the minimum and maximum elements.  
12 a_min = a_max = a[0]  
13 i_min = i_max = 0  
14 for i, item in enumerate(a):  
15     if item < a_min:  
16         i_min, a_min = i, item  
17     if item >= a_max:  
18         i_max, a_max = i, item  
19  
20     # Check indexes and swap them.  
21     if i_min > i_max:  
22         i_min, i_max = i_max, i_min  
23  
24     # Count the number of positive elements.  
25     count = 0  
26     for item in a[i_min + 1:i_max]:  
27         if item > 0:  
28             count += 1  
29     print(count)  
if __name__ == '__main__': for i, item in enumerate(a) > if item >= a_max
```

The bottom panel shows the 'Python Console' with the following output:

```
D:\anaconda\python.exe "C:/Program Files/JetBrains/PyCharm Community Edition 2023.1.2/plugins/python-ce/helpers/pydev/pydevconsole.py" --mode=client --host  
Couldn't connect to console process.  
Process finished with exit code -1
```

The status bar at the bottom indicates the current file is 'ex2.py', the encoding is 'UTF-8', and the Python version is '3.10'.

Рис. 4. Программа и ее результат.

5. Индивидуальное задание №1.

Вариант 3 (по списку группы). Условие задания:

Ввести список A из 10 элементов, найти наименьший элемент и переставить его с последним элементом. Преобразованный массив вывести.

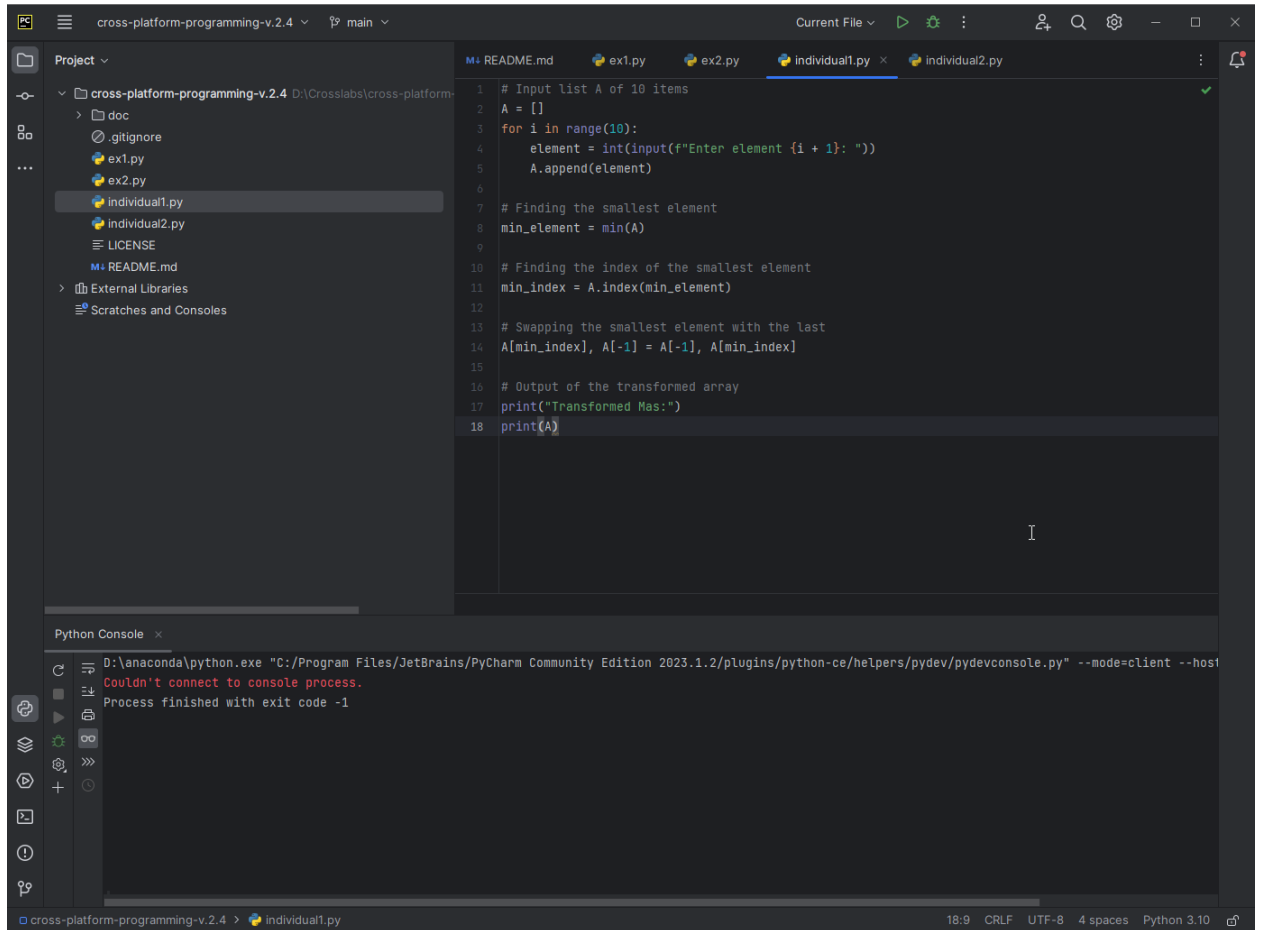
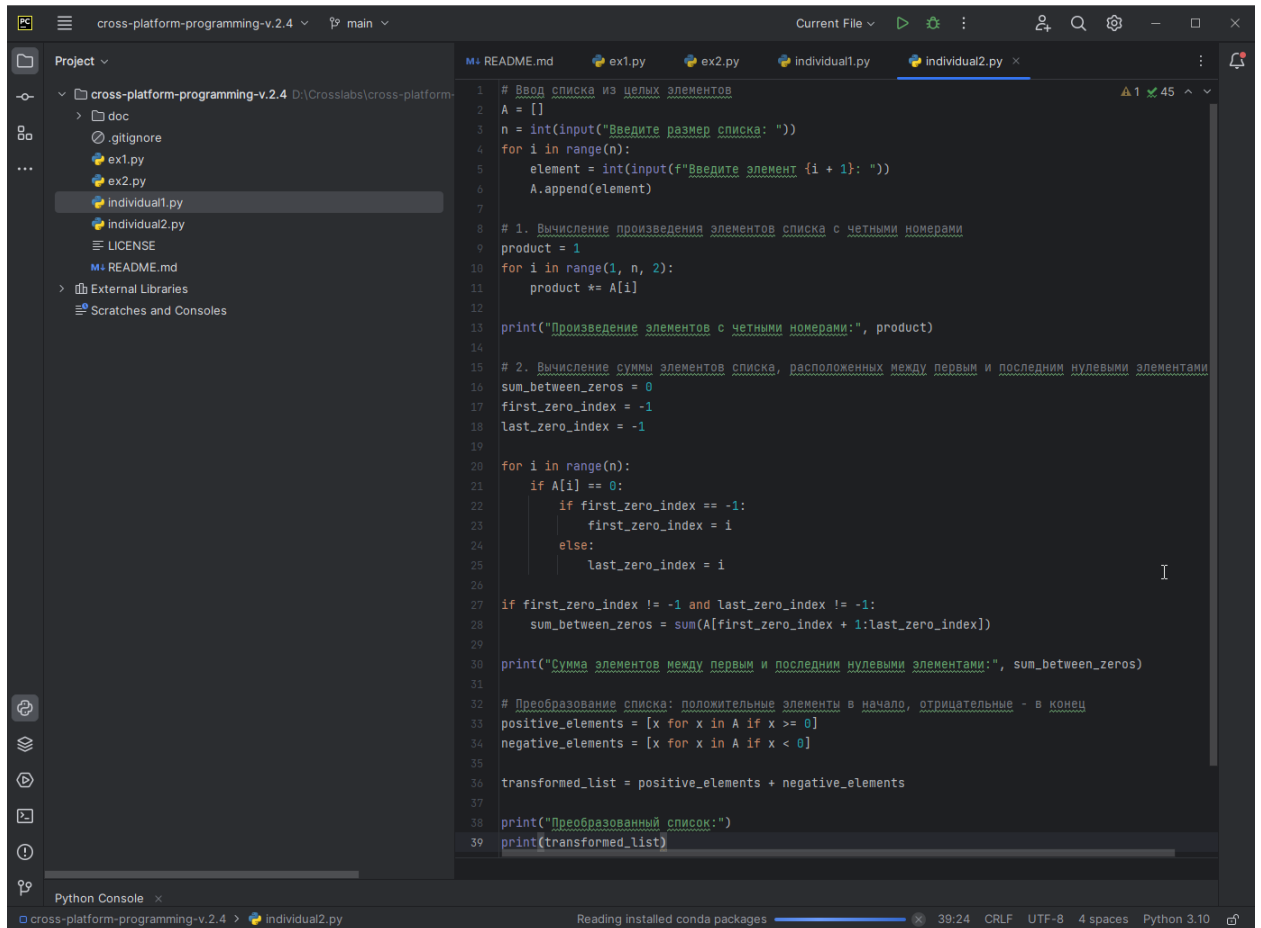


Рис. 9. Программа и ее результат.

6. Индивидуальное задание №2.

Вариант 3 (по списку группы). Условие задания:

В списке, состоящем из целых элементов, вычислить: 1. произведение элементов списка с четными номерами; 2. сумму элементов списка, расположенных между первым и последним нулевыми элементами.



The screenshot shows a code editor with a project named 'cross-platform-programming-v.2.4'. The file explorer on the left shows a directory structure with files like 'ex1.py', 'ex2.py', 'individual1.py', and 'individual2.py'. The main editor window displays the code for 'individual2.py'. The code is as follows:

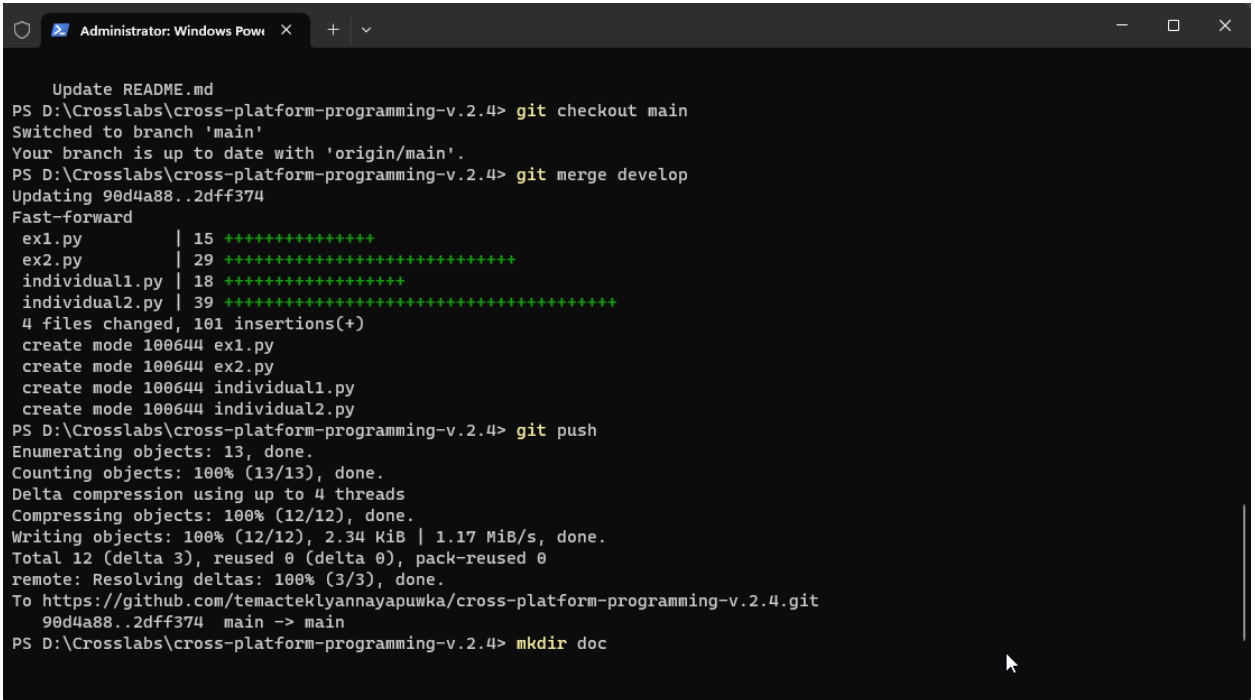
```
1 # Ввод списка из целых элементов
2 A = []
3 n = int(input("Введите размер списка: "))
4 for i in range(n):
5     element = int(input(f"Введите элемент {i + 1}: "))
6     A.append(element)
7
8 # 1. Вычисление произведения элементов списка с четными номерами
9 product = 1
10 for i in range(1, n, 2):
11     product *= A[i]
12
13 print("Произведение элементов с четными номерами:", product)
14
15 # 2. Вычисление суммы элементов списка, расположенных между первым и последним нулевыми элементами
16 sum_between_zeros = 0
17 first_zero_index = -1
18 last_zero_index = -1
19
20 for i in range(n):
21     if A[i] == 0:
22         if first_zero_index == -1:
23             first_zero_index = i
24         else:
25             last_zero_index = i
26
27 if first_zero_index != -1 and last_zero_index != -1:
28     sum_between_zeros = sum(A[first_zero_index + 1:last_zero_index])
29
30 print("Сумма элементов между первым и последним нулевыми элементами:", sum_between_zeros)
31
32 # Преобразование списка: положительные элементы в начало, отрицательные - в конец
33 positive_elements = [x for x in A if x >= 0]
34 negative_elements = [x for x in A if x < 0]
35
36 transformed_list = positive_elements + negative_elements
37
38 print("Преобразованный список:")
39 print(transformed_list)
```

The bottom of the image shows the Python Console with the output of the program. The output is:

```
cross-platform-programming-v.2.4 > individual2.py
Введите размер списка: 10
Введите элемент 1: 1
Введите элемент 2: 2
Введите элемент 3: 3
Введите элемент 4: 4
Введите элемент 5: 5
Введите элемент 6: 6
Введите элемент 7: 7
Введите элемент 8: 8
Введите элемент 9: 9
Введите элемент 10: 10
Произведение элементов с четными номерами: 10368
Сумма элементов между первым и последним нулевыми элементами: 0
Преобразованный список:
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
```

Рис. 10. Программа и ее результат.

7. Слил ветку develop с веткой main и отправил на удаленный сервер – Github.



```
Update README.md
PS D:\Crosslabs\cross-platform-programming-v.2.4> git checkout main
Switched to branch 'main'
Your branch is up to date with 'origin/main'.
PS D:\Crosslabs\cross-platform-programming-v.2.4> git merge develop
Updating 90d4a88..2dff374
Fast-forward
  ex1.py      | 15 ++++++
  ex2.py      | 29 ++++++
  individual1.py | 18 ++++++
  individual2.py | 39 ++++++
  4 files changed, 101 insertions(+)
  create mode 100644 ex1.py
  create mode 100644 ex2.py
  create mode 100644 individual1.py
  create mode 100644 individual2.py
PS D:\Crosslabs\cross-platform-programming-v.2.4> git push
Enumerating objects: 13, done.
Counting objects: 100% (13/13), done.
Delta compression using up to 4 threads
Compressing objects: 100% (12/12), done.
Writing objects: 100% (12/12), 2.34 KiB | 1.17 MiB/s, done.
Total 12 (delta 3), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (3/3), done.
To https://github.com/temacteklyannayapuwka/cross-platform-programming-v.2.4.git
  90d4a88..2dff374  main -> main
PS D:\Crosslabs\cross-platform-programming-v.2.4> mkdir doc
```

Рис. 12. Слияние веток.

Ответы на контрольные вопросы:

Что такое списки в языке Python? Список (list) – это структура данных для хранения объектов различных типов. В нем можно хранить объекты различных типов. Размер списка не статичен, его можно изменять. Список по своей природе является изменяемым типом данных. Переменная, определяемая как список, содержит ссылку на структуру в памяти, которая в свою очередь хранит на какие-либо другие объекты или структуры.

2. Как осуществляется создание списка в Python? Для создания списка нужно заключить элементы в квадратные скобки.

3. Как организовано хранение списков в оперативной памяти? При создании списка в памяти резервируется область, которую можно условно назвать некоторым “контейнером”, в котором хранятся ссылки другие элементы данных в памяти. В отличие от таких типов данных число или строка, содержимое “контейнера” списка можно менять.

4. Каким образом можно перебрать все элементы списка? Читать элементы списка можно с помощью следующего цикла:
`my_list = ['один', 'два', 'три', 'четыре', 'пять']`
`for elem in my_list: print(elem)`

5. Какие существуют арифметические операции со списками? Для объединения списков можно использовать оператор сложения (`+`). Список можно повторить с помощью оператора умножения (`*`).

6. Как проверить есть ли элемент в списке? Для того, чтобы проверить, есть ли заданный элемент в списке Python необходимо использовать оператор `in`.

7. Как определить число вхождений заданного элемента в списке? Метод `count` можно использовать для определения числа сколько раз данный элемент встречается в списке.

8. Как осуществляется добавление (вставка) элемента в список? Метод `append` можно использовать для добавления элемента в список. Метод `insert` можно использовать, чтобы вставить элемент в список.

9. Как выполнить сортировку списка? Для сортировки списка нужно использовать метод `sort`. Для сортировки списка в порядке убывания необходимо вызвать метод `sort` с аргументом `reverse=True`.

10. Как удалить один или несколько элементов из списка? Удалить элемент можно, написав его индекс в методе `pop`. Если не указывать индекс, то функция удалит последний элемент. Элемент можно удалить с помощью метода `remove`. Оператор `del` можно использовать для тех же целей. Можно удалить несколько элементов с помощью оператора среза. Можно удалить все элементы из списка с помощью метода `clear`.

11. Что такое списковое включение и как с его помощью осуществлять обработку списков? List Comprehensions чаще всего на русский язык переводят как абстракция списков или списковое включение, является частью синтаксиса языка, которая предоставляет простой способ построения списков. В языке Python есть две очень мощные функции для работы с коллекциями: `map` и `filter`. Они позволяют использовать функциональный стиль

программирования, не прибегая к помощи циклов, для работы с такими типами как list, tuple, set, dict и т.п. Списковое включение позволяет обойтись без этих функций.

12. Какие существуют функции агрегации для работы со списками? Для работы со списками Python предоставляет следующие функции: 1. len(L) - получить число элементов в списке L 2. min(L) - получить минимальный элемент списка L 3. max(L) - получить максимальный элемент списка L 4. sum(L) - получить сумму элементов списка L, если список L содержит только числовые значения.

13. Как создать копию списка? `copy.copy(x)`

14. Самостоятельно изучите функцию sorted языка Python. В чем ее отличие от метода sort списков? Функция sorted() в Python возвращает отсортированный список из элементов в итерируемом объекте. list.sort() на 13% быстрее, чем sorted().

15. Самостоятельно изучите функцию sorted языка Python. В чем ее отличие от метода sort списков? Функция sort() очень похожа на sorted(), но в отличие от sorted она ничего не возвращает и не вносит изменений в исходную последовательность. Более того, sort() является методом класса list и может использоваться только со списками. Синтаксис: List_name.sort(key, reverse=False) Параметры: ключ: Функция, которая служит ключом для сравнения сортировки. реверс: Если true, то список сортируется в порядке убывания.

Вывод: в ходе выполнения лабораторной работы приобретены навыки по работе со списками при написании программ с помощью языка программирования Python версии 3.x.