

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 ЛАБОРАТОРНОЙ РАБОТЕ №2.4
дисциплины
«Основы кроссплатформенного программирования»

Выполнил:
Наумов Никита Викторович 1 курс,
группа ИТС-б-о-22-1,
11.03.02 «Инфокоммуникационные
технологии и системы связи»,
направленность (профиль)
«Инфокоммуникационные системы и
сети», заочная форма обучения

(подпись)

Проверил:
Воронкин Р. А., доцент кафедры
инфокоммуникаций

(подпись)

Отчет защищен с оценкой _____ Дата защиты _____

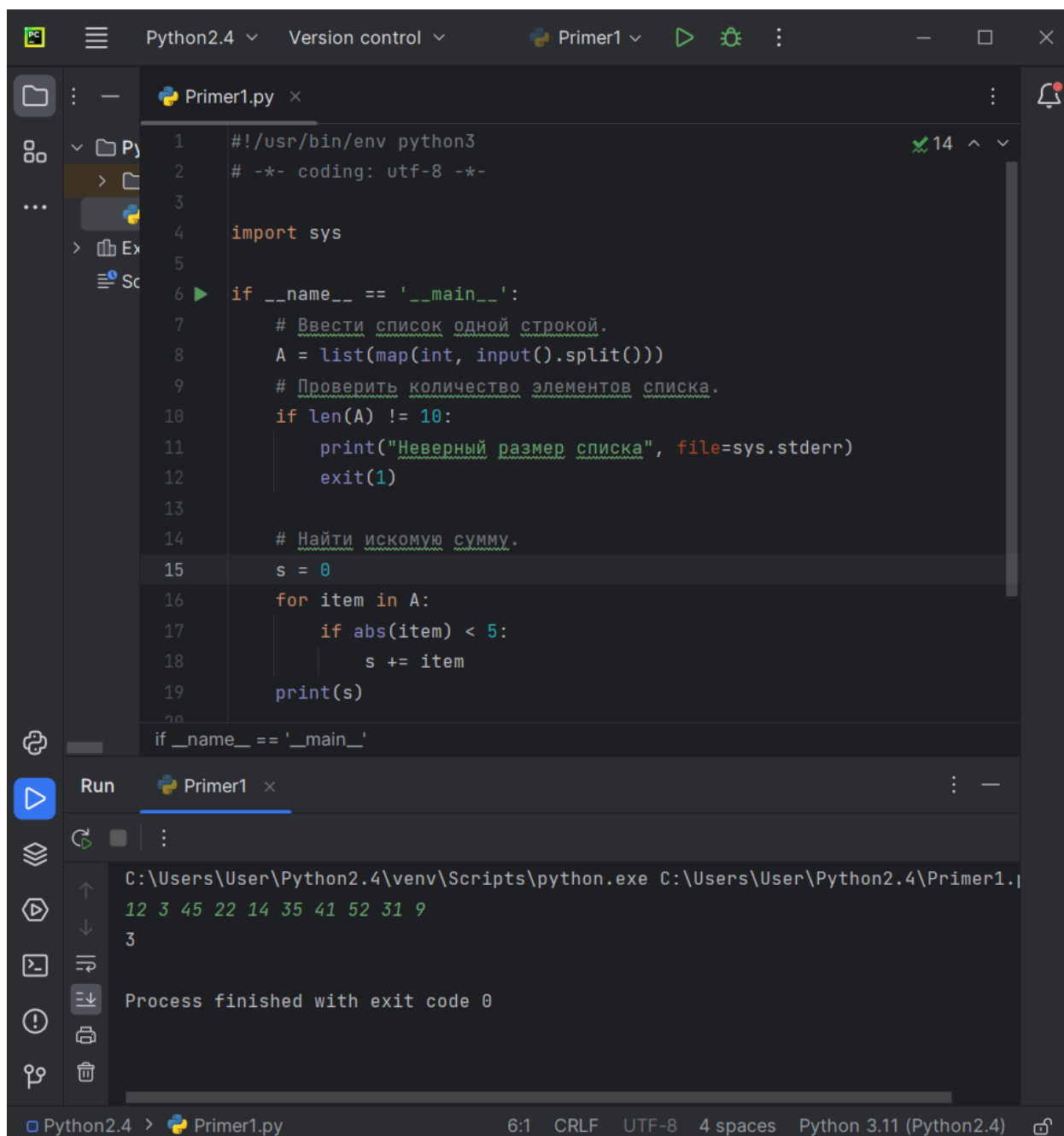
Ставрополь, 2023 г.

Тема: Работа со списками в языке Python.

Цель: Приобретение навыков по работе со списками при написании программ с помощью языка программирования Python версии 3.x.

Ход работы:

Задание 1.



```
1  #!/usr/bin/env python3
2  # -*- coding: utf-8 -*-
3
4  import sys
5
6  if __name__ == '__main__':
7      # Ввести список одной строкой.
8      A = list(map(int, input().split()))
9      # Проверить количество элементов списка.
10     if len(A) != 10:
11         print("Неверный размер списка", file=sys.stderr)
12         exit(1)
13
14     # Найти искомую сумму.
15     s = 0
16     for item in A:
17         if abs(item) < 5:
18             s += item
19     print(s)
20
21 if __name__ == '__main__':
```

Run Primer1 x

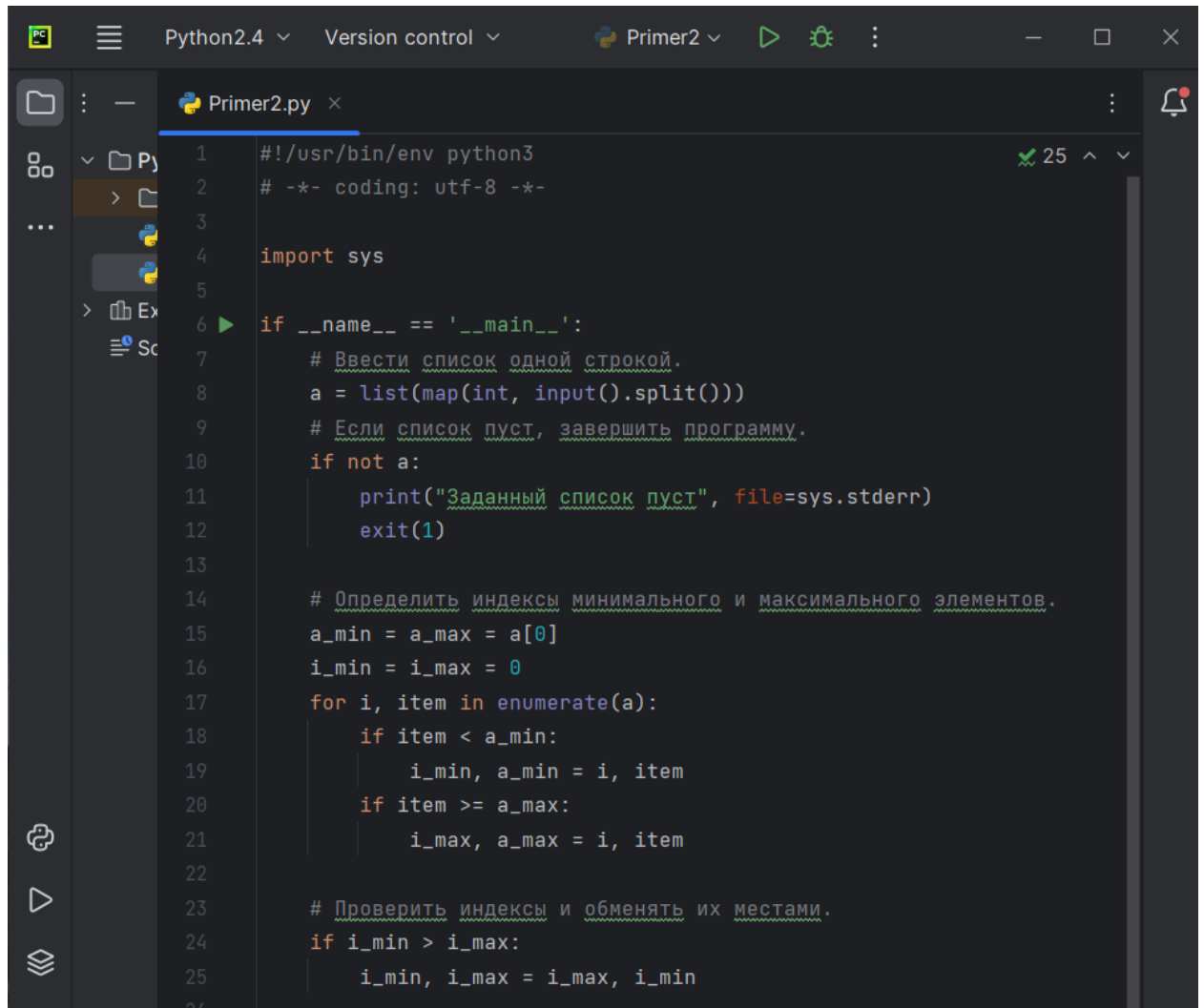
C:\Users\User\Python2.4\env\Scripts\python.exe C:\Users\User\Python2.4\Primer1.py
12 3 45 22 14 35 41 52 31 9
3

Process finished with exit code 0

Python2.4 > Primer1.py 6:1 CRLF UTF-8 4 spaces Python 3.11 (Python2.4)

Рисунок 1 – Пример 1.

Пример 2.



```
1  #!/usr/bin/env python3
2  # -*- coding: utf-8 -*-
3
4  import sys
5
6  if __name__ == '__main__':
7      # Ввести список одной строкой.
8      a = list(map(int, input().split()))
9      # Если список пуст, завершить программу.
10     if not a:
11         print("Заданный список пуст", file=sys.stderr)
12         exit(1)
13
14     # Определить индексы минимального и максимального элементов.
15     a_min = a_max = a[0]
16     i_min = i_max = 0
17     for i, item in enumerate(a):
18         if item < a_min:
19             i_min, a_min = i, item
20         if item >= a_max:
21             i_max, a_max = i, item
22
23     # Проверить индексы и обменять их местами.
24     if i_min > i_max:
25         i_min, i_max = i_max, i_min
26
```

Рисунок 2 – Пример 2(начало).

The image shows a screenshot of a Python IDE. The main editor window displays a file named `Primer2.py` with the following code:

```
21         i_max, a_max = i, item
22
23         # Проверить индексы и обменять их местами.
24         if i_min > i_max:
25             i_min, i_max = i_max, i_min
26
27         # Посчитать количество положительных элементов.
28         count = 0
29         for item in a[i_min+1:i_max]:
30             if item > 0:
31                 count += 1
32         print(count)
33
34 if __name__ == '__main__':
```

Below the editor is a `Run` panel for `Primer2`. It shows the command executed: `C:\Users\User\Python2.4\venv\Scripts\python.exe C:\Users\User\Python2.4\Primer2.py`. The output of the program is displayed as:

```
-4 54 -21 12 42 -63
2
```

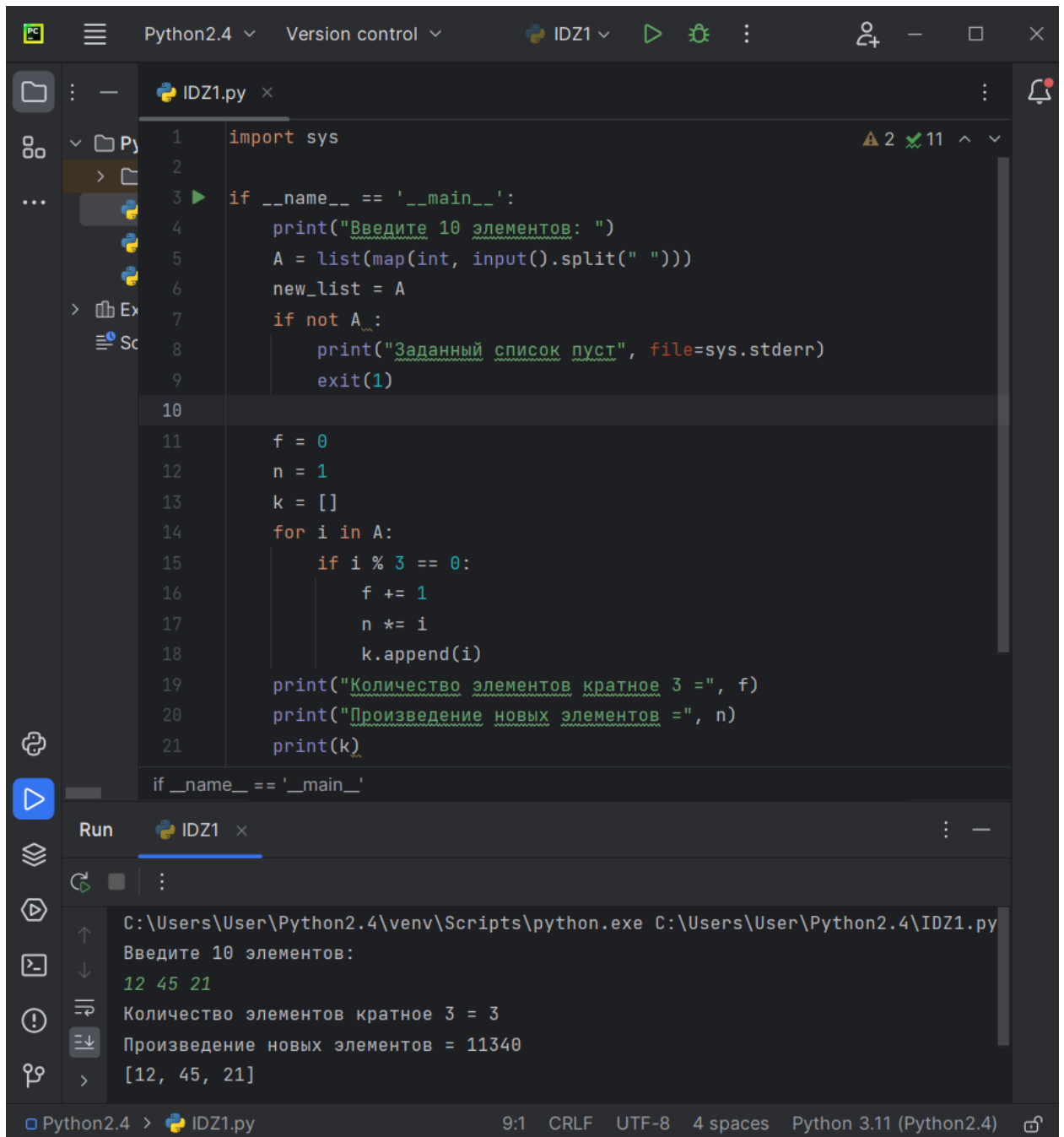
At the bottom of the Run panel, it states: `Process finished with exit code 0`. The status bar at the bottom indicates the file is `Primer2.py`, the encoding is `UTF-8`, and the Python version is `Python 3.11 (Python2.4)`.

Рисунок 3 – Пример 2(конец).

Индивидуальные задания:

Вариант №10.

Задание 1.



The screenshot shows a code editor with a Python script named IDZ1.py. The script prompts the user to enter 10 elements, processes them, and prints the results. The output window shows the execution of the script, including the user input and the calculated values.

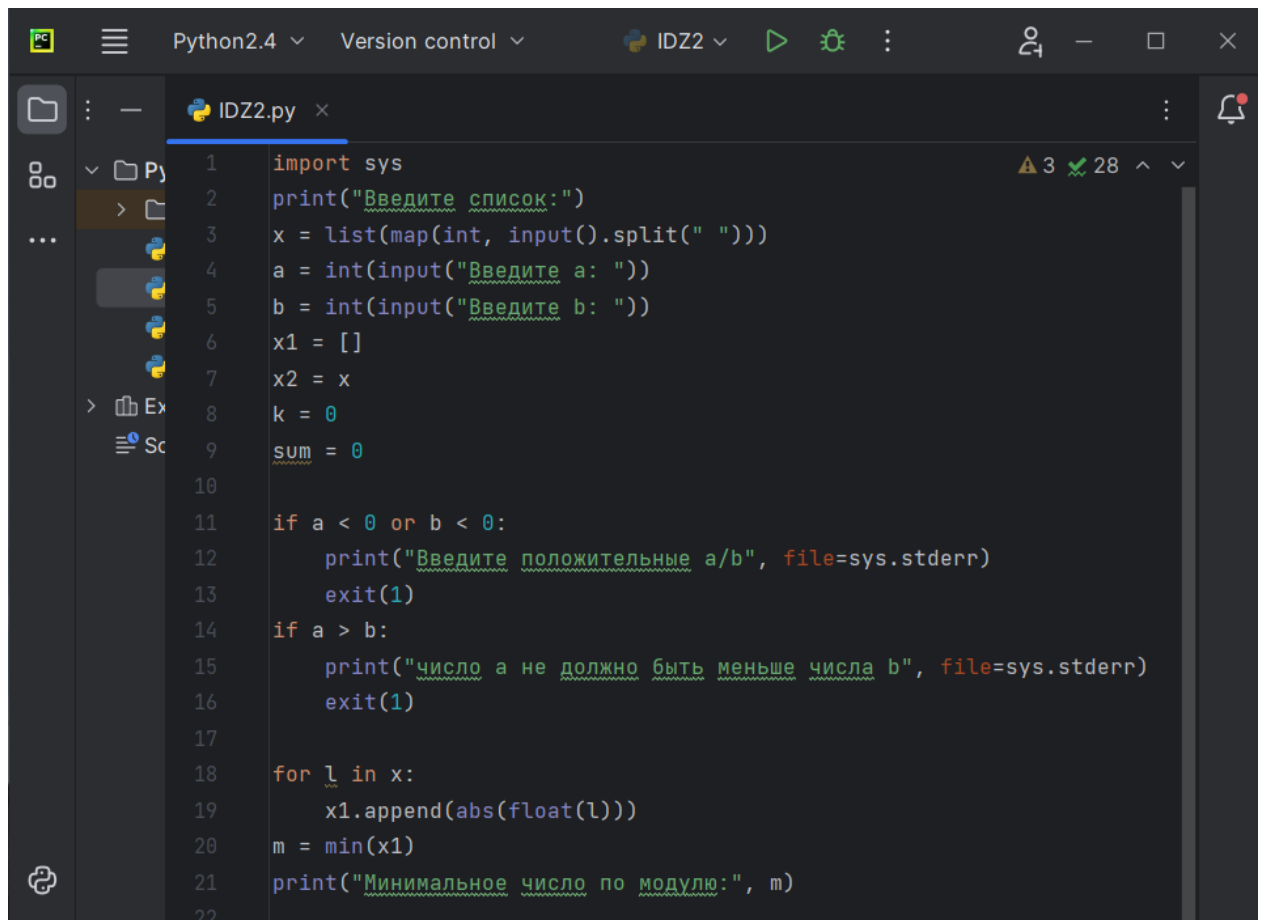
```
1 import sys
2
3 if __name__ == '__main__':
4     print("Введите 10 элементов: ")
5     A = list(map(int, input().split(" ")))
6     new_list = A
7     if not A:
8         print("Заданный список пуст", file=sys.stderr)
9         exit(1)
10
11     f = 0
12     n = 1
13     k = []
14     for i in A:
15         if i % 3 == 0:
16             f += 1
17             n *= i
18             k.append(i)
19     print("Количество элементов кратное 3 =", f)
20     print("Произведение новых элементов =", n)
21     print(k)
```

Run IDZ1

```
C:\Users\User\Python2.4\venv\Scripts\python.exe C:\Users\User\Python2.4\IDZ1.py
Введите 10 элементов:
12 45 21
Количество элементов кратное 3 = 3
Произведение новых элементов = 11340
[12, 45, 21]
```

Рисунок 4 – ИДЗ 1.

Задание 2.



The screenshot shows an IDE window with a Python file named IDZ2.py. The code is as follows:

```
1 import sys
2 print("Введите список:")
3 x = list(map(int, input().split(" ")))
4 a = int(input("Введите a: "))
5 b = int(input("Введите b: "))
6 x1 = []
7 x2 = x
8 k = 0
9 sum = 0
10
11 if a < 0 or b < 0:
12     print("Введите положительные a/b", file=sys.stderr)
13     exit(1)
14 if a > b:
15     print("число a не должно быть меньше числа b", file=sys.stderr)
16     exit(1)
17
18 for l in x:
19     x1.append(abs(float(l)))
20 m = min(x1)
21 print("Минимальное число по модулю:", m)
```

The IDE interface includes a top toolbar with icons for Python, menu, version control, and execution. The left sidebar shows a file explorer with a project structure. The right sidebar shows a notification bell icon. The code is written in a dark-themed editor with syntax highlighting.

Рисунок 5 – ИДЗ 2(начало).

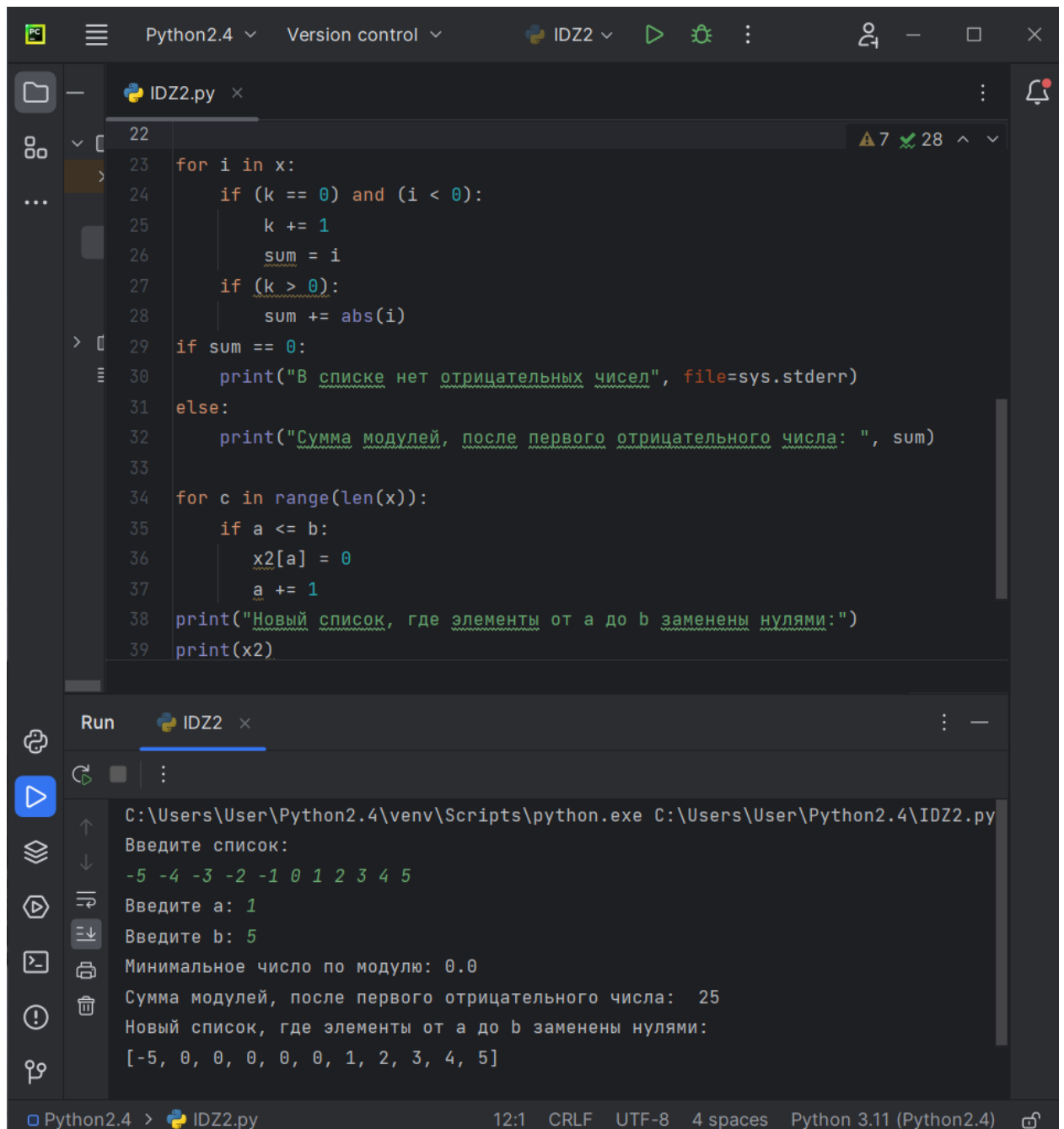


Рисунок 6 – ИДЗ 2(конец).

Контрольные вопросы:

1. 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 на другие элементы данных в памяти. В отличие от таких типов данных, как 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.