

РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ

Факультет физико-математических и естественных наук

Кафедра теории вероятностей и кибербезопасности

ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ № 1

дисциплина: Компьютерные науки и технологии программирования

Студент: Захаров Александр Петрович

Группа: НММбд-02-23

Преподаватель: Бегишев В.О.

МОСКВА

2024 г.

Цель работы:

1. Познакомиться со средой разработки Python. Изучить основные типы данных, команды ввода и вывода данных.
2. Познакомиться со структурой ветвление (if, if-else, if-elif-else).
3. Познакомиться с циклическими конструкциями

Выполнение работы

Задание 1.

Условие:

Напишем программу, которая запрашивала бы у пользователя:

- ФИО ("Ваши фамилия, имя, отчество?")
- возраст ("Сколько Вам лет?")
- место жительства ("Где вы живете?")

После этого выводила бы три строки:

"Ваше имя"

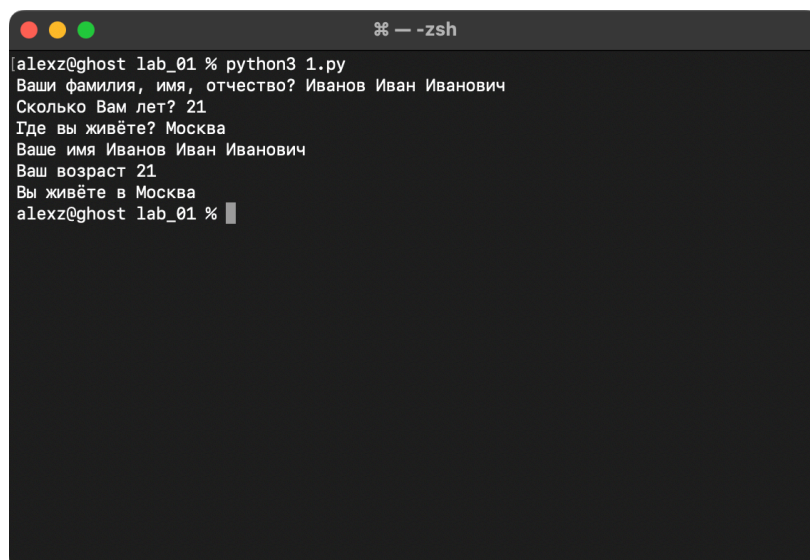
"Ваш возраст"

"Вы живете в"

Листинг программы:

```
1. imya = input('Ваши фамилия, имя, отчество? ')
2. vozrast = input('Сколько Вам лет? ')
3. gorod = input('Где вы живёте? ')
4. print("Ваше имя", imya)
5. print("Ваш возраст", vozrast)
6. print("Вы живёте в", gorod)
```

Описание результатов (численные значения, графики, скриншоты), анализ программы
(тезисно, что сделано в работе).



```
alexz@ghost lab_01 % python3 1.py
Ваши фамилия, имя, отчество? Иванов Иван Иванович
Сколько Вам лет? 21
Где вы живёте? Москва
Ваше имя Иванов Иван Иванович
Ваш возраст 21
Вы живёте в Москва
alexz@ghost lab_01 %
```

Тезисное описание программы:

- Последовательно запрашиваются фамилия, имя и отчество
- Выводятся на экран

Задание 2.

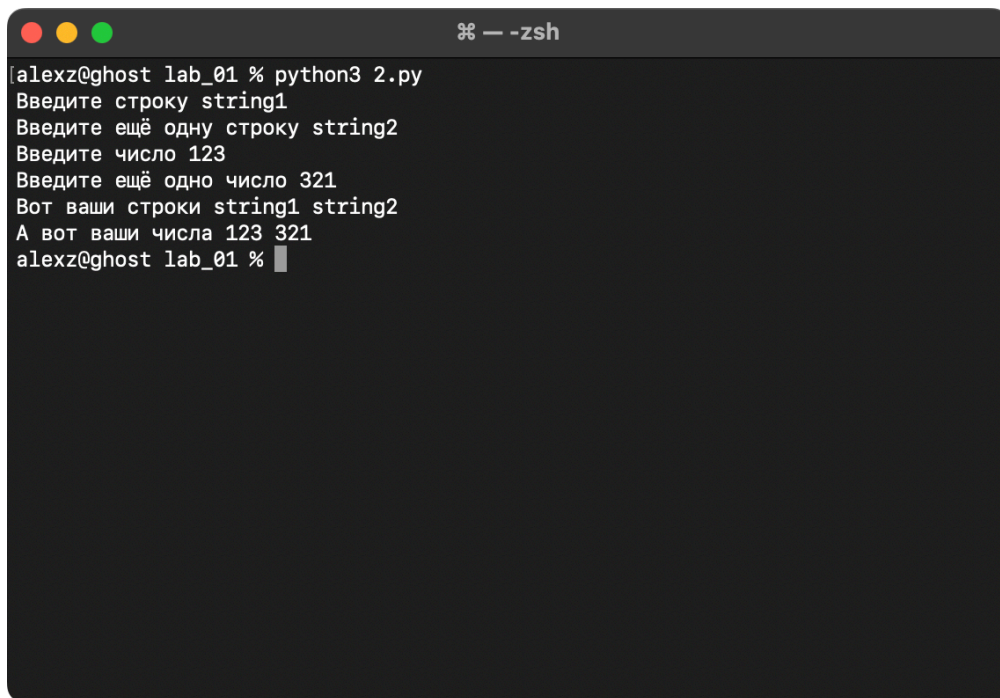
Условие:

Запрос у пользователя пару чисел и строк и сохранение в переменные далее вывод на экран.

Листинг:

```
1. Str_in_1 = input("Введите строку ")
2. str_in_2 = input("Введите ещё одну строку ")
3. int_in_1 = int(input("Введите число "))
4. int_in_2 = int(input("Введите ещё одно число "))
5.
6. print("Вот ваши строки", str_in_1, str_in_2)
7. print("А вот ваши числа", int_in_1, int_in_2)
```

Описание результатов (численные значения, графики, скриншоты), анализ программы
(тезисно, что сделано в работе).



```
alexz@ghost lab_01 % python3 2.py
Введите строку string1
Введите ещё одну строку string2
Введите число 123
Введите ещё одно число 321
Вот ваши строки string1 string2
А вот ваши числа 123 321
alexz@ghost lab_01 %
```

Тезисное описание программы:

- Запрашиваются две строки и два числа
- Выводятся на экран

Задание 3.

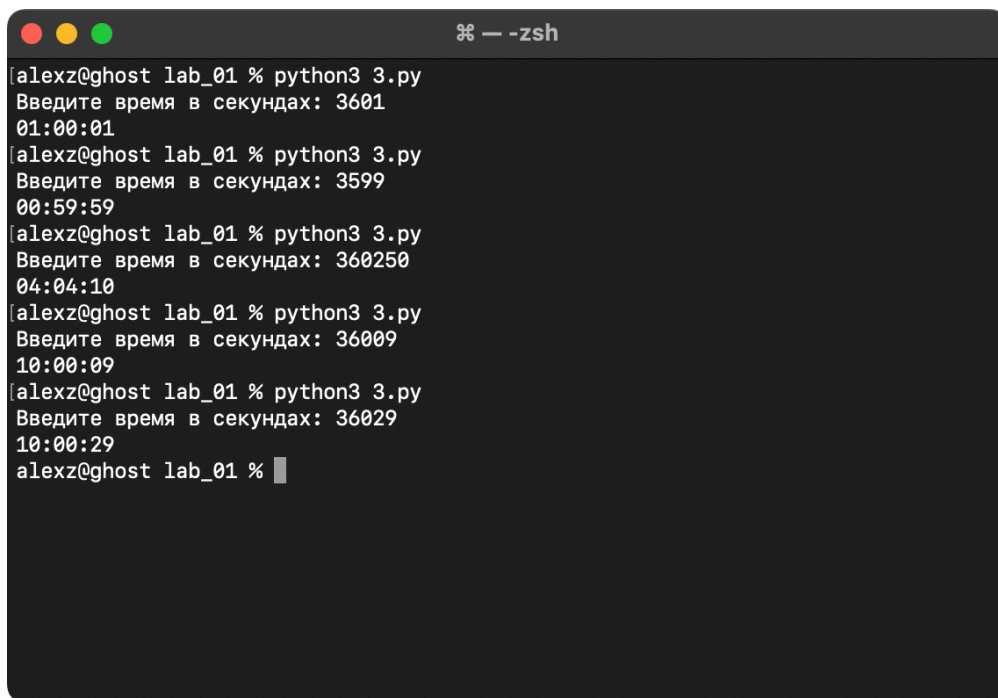
Условие:

Вы вводите время в секундах. Вам необходимо перевести время в часы, минуты и секунды и далее выводите в формате – чч:мм:сс. Применяйте форматирование строк.

Листинг:

```
1. vremya = int(input("Введите время в секундах: "))
2. chasov = vremya // 3600
3. minut = (vremya - (chasov * 3600)) // 60
4. sekund = vremya - (chasov * 3600) - (minut * 60)
5. chasov = chasov % 24
6.
7. print(f"{chasov:0>2}:{minut:0>2}:{sekund:0>2}")
```

Описание результатов (численные значения, графики, скриншоты), анализ программы (тезисно, что сделано в работе).



```
alexz@ghost lab_01 % python3 3.py
Введите время в секундах: 3601
01:00:01
alexz@ghost lab_01 % python3 3.py
Введите время в секундах: 3599
00:59:59
alexz@ghost lab_01 % python3 3.py
Введите время в секундах: 360250
04:04:10
alexz@ghost lab_01 % python3 3.py
Введите время в секундах: 36009
10:00:09
alexz@ghost lab_01 % python3 3.py
Введите время в секундах: 36029
10:00:29
alexz@ghost lab_01 %
```

Тезисное описание программы:

- Вводится число секунд
- Из секунд выделяется целое число часов и записывается в переменную **chasov**
- Вычисляется целое число минут и записывается в переменную **minut**
- Вычисляется целое число секунд и записывается в переменную **sekund**
- Берётся модуль от количества часов, ведь их не может быть более 24 в сутках
- Ответ выводится на экран в виде форматированной строки в соответствии с заданием

Задание 4.

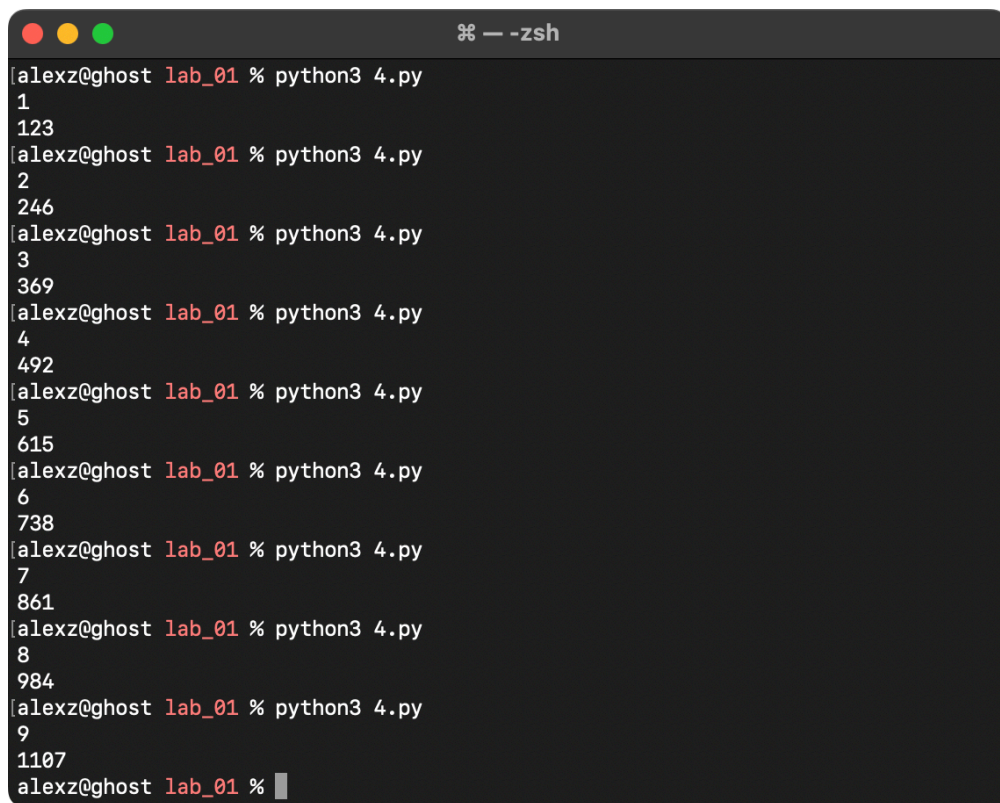
Условие:

Ввести число n с клавиатуры ($1 \leq n \leq 9, n \in \mathbb{N}$) найти сумму чисел $n + nn + nnn$.

Листинг:

```
1. n = int(input())
2. print(n + n*11 + n * 111)
```

Описание результатов (численные значения, графики, скриншоты), анализ программы
(тезисно, что сделано в работе).



```
alexz@ghost lab_01 % python3 4.py
1
123
alexz@ghost lab_01 % python3 4.py
2
246
alexz@ghost lab_01 % python3 4.py
3
369
alexz@ghost lab_01 % python3 4.py
4
492
alexz@ghost lab_01 % python3 4.py
5
615
alexz@ghost lab_01 % python3 4.py
6
738
alexz@ghost lab_01 % python3 4.py
7
861
alexz@ghost lab_01 % python3 4.py
8
984
alexz@ghost lab_01 % python3 4.py
9
1107
alexz@ghost lab_01 %
```

Тезисное описание программы:

- Запрошено число с клавиатуры
- Вывод числа $n + nn + nnn$ через умножение на 11 и 111 в последних двух случаях соответственно

Задание 5.

Условие:

Введите целое положительное число. Попробуйте найти самую большую цифру в числе. Для решения примените цикл `while` и арифметические операции

Листинги:

Алгоритм А.

```
1. a = input()
2. max = 0
3. for i in a:
4.     if int(i) > max:
5.         max = int(i)
6. print(max)
```

Описание результатов (численные значения, графики, скриншоты), анализ программы (*тезисно, что сделано в работе*).

Тезисное описание программы:

- Требуемый алгоритм реализован через поиск в строке – а
- С клавиатур вводится строка в переменную a
- С помощью цикла for переменная i последовательно проходит по символам в этой строке, преобразовывает их в число, сравнивает с максимумом
- Если максимум оказался меньше числа в строке, то переменная max сохраняет новый максимум равный этому числу
- Вывод максимума на экран

Алгоритм Б.

```
1. a = int(input())
2. max = 0
3. while a > 0:
4.     i = a % 10
5.     a = a // 10
6.     if i > max:
7.         max = i
8. print(max)
```

Описание результатов (численные значения, графики, скриншоты), анализ программы (*тезисно, что сделано в работе*).

Тезисное описание программы:

[illegible][illegible]

- Требуемый алгоритм реализован через деление числа – a
- С клавиатур вводится число в переменную a
- С помощью цикла while от числа a отделяются цифры, начиная с конца
- Если максимум оказался меньше текущей отделённой цифры, то переменная max сохраняет новый максимум равный этой цифре
- Вывод максимума на экран

Задание 6.

Условие:

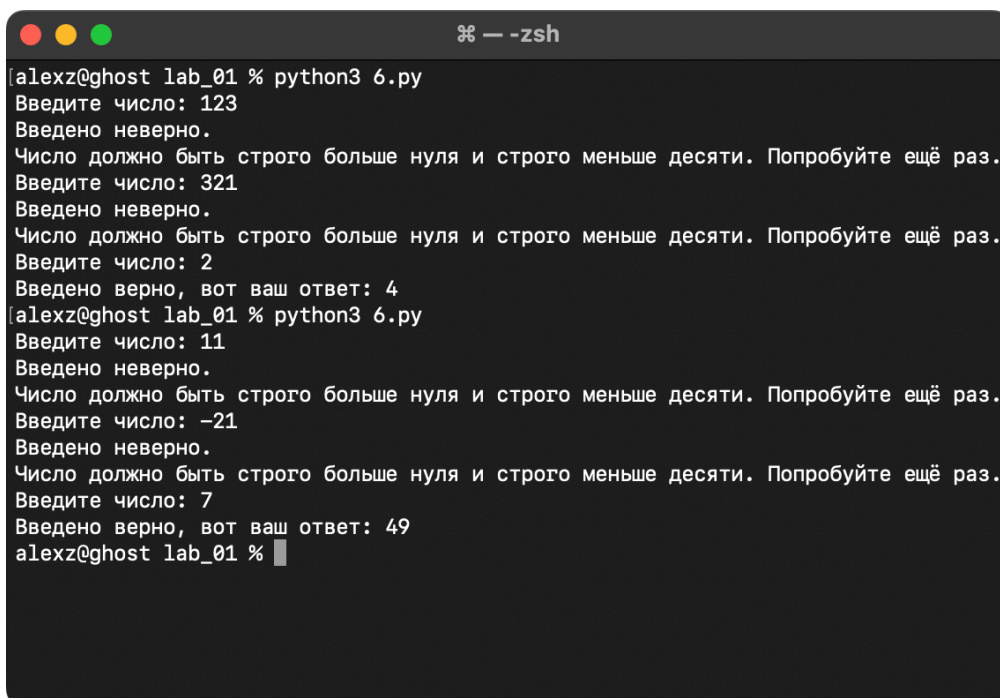
Используя цикл, запрашивайте у пользователя число, пока оно не станет больше 0, но меньше 10. После того, как пользователь введет корректное число, возведите его в степень 2 и выведите на экран.

Например, пользователь вводит число 123, вы сообщаете ему, что число неверное, говорите о диапазоне допустимых и просите ввести заново.

Листинг:

```
1. a = 11
2. while (a >= 10) or (a <= 0):
3.     a = int(input("Введите число: "))
4.     if (a >= 10) or (a <= 0):
5.         print("Введено неверно.")
6.         print(
7.             "Число должно быть строго больше нуля и строго меньше десяти.
            Попробуйте ещё раз."
8.         )
9. print("Введено верно, вот ваш ответ:", a**2)
```

Описание результатов (численные значения, графики, скриншоты), анализ программы (тезисно, что сделано в работе).



```
alexz@ghost lab_01 % python3 6.py
Введите число: 123
Введено неверно.
Число должно быть строго больше нуля и строго меньше десяти. Попробуйте ещё раз.
Введите число: 321
Введено неверно.
Число должно быть строго больше нуля и строго меньше десяти. Попробуйте ещё раз.
Введите число: 2
Введено верно, вот ваш ответ: 4
alexz@ghost lab_01 % python3 6.py
Введите число: 11
Введено неверно.
Число должно быть строго больше нуля и строго меньше десяти. Попробуйте ещё раз.
Введите число: -21
Введено неверно.
Число должно быть строго больше нуля и строго меньше десяти. Попробуйте ещё раз.
Введите число: 7
Введено верно, вот ваш ответ: 49
alexz@ghost lab_01 %
```

Тезисное описание программы:

- С помощью цикла while с клавиатуры запрашивается число и записывается в переменную
- Условным оператором if число проверяется на удовлетворение данным в задаче ограничениям
- Как только оказывается введено нужное число, цикл завершается и выводится это число в квадрате на экран

Задание 7. Условие:

Создайте программу “Медицинская анкета”, где вы запросите у пользователя следующие данные: имя, фамилия, возраст и вес. Выведите результат согласно которому:

- Пациент в хорошем состоянии, если ему до 30 лет и вес от 50 и до 120 кг,
- Пациенту требуется заняться собой, если ему более 30 и вес меньше 50 или больше 120 кг
- Пациенту требуется врачебный осмотр, если ему более 40 и вес менее 50 или больше 120 кг.
- Все остальные варианты вы можете обработать на ваш вкус и полет фантазии.

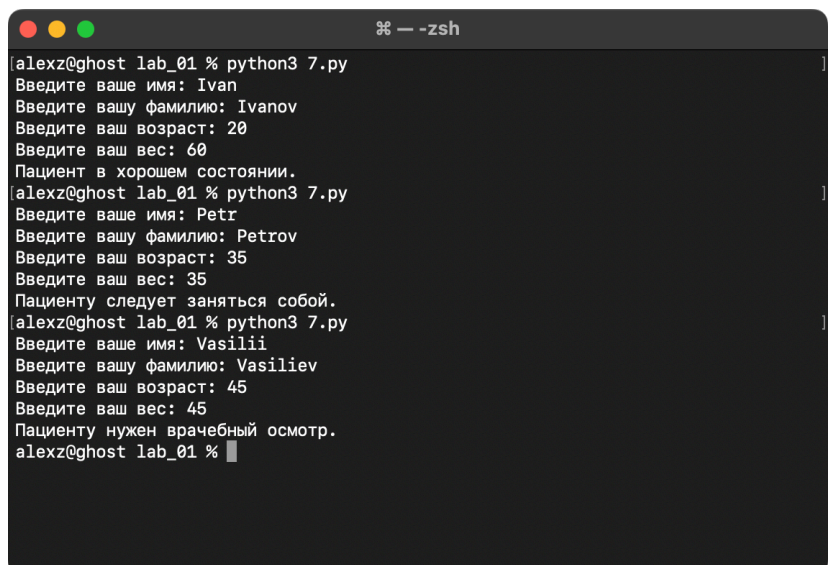
Листинг:

```
1. imya = input("Введите ваше имя: ")
2. familiya = input("Введите вашу фамилию: ")
3. vozrast = int(input("Введите ваш возраст: "))
4. ves = int(input("Введите ваш вес: "))
5.
6. if (vozrast < 30) and (50 <= ves <= 120):
7.     print("Пациент в хорошем состоянии.")
8. elif (vozrast > 30) and (vozrast <= 40) and ((ves < 50) or (ves > 120)):
9.     print("Пациенту следует заняться собой.")
10. elif (vozrast > 40) and ((ves < 50) or (ves > 120)):
11.     print("Пациенту нужен врачебный осмотр.")
12. else:
13.     print("Я СИГМА КРУТОЙ Я СИГМА СИГМА СИГМА СИГМА")
```

Описание результатов (численные значения, графики, скриншоты), анализ программы (тезисно, что сделано в работе).

Тезисное описание программы:

- Запрашиваются данные пациента и записываются в переменные, как требуется в задании,
- С помощью условного оператора выводятся необходимые по условию задачи ответы.



```
% — -zsh
alexz@ghost lab_01 % python3 7.py
Введите ваше имя: Ivan
Введите вашу фамилию: Ivanov
Введите ваш возраст: 20
Введите ваш вес: 60
Пациент в хорошем состоянии.
alexz@ghost lab_01 % python3 7.py
Введите ваше имя: Petr
Введите вашу фамилию: Petrov
Введите ваш возраст: 35
Введите ваш вес: 35
Пациенту следует заняться собой.
alexz@ghost lab_01 % python3 7.py
Введите ваше имя: Vasilii
Введите вашу фамилию: Vasiliev
Введите ваш возраст: 45
Введите ваш вес: 45
Пациенту нужен врачебный осмотр.
alexz@ghost lab_01 %
```


- Вводятся исходные точки – в переменную `x` результат в первый день, а в переменную `y` – целевой результат.
- Объявляется переменная `iterator` для подсчёта количества дней (выполнений цикла)
- Цикл `while` выполняется до тех пор, пока значение результата в переменной `x` не превысит цели в `y`.
- На экран выводится количество дней, которое на это потребуется, через переменную `iterator`

Заключение.

В ходе лабораторной работы я успешно познакомился с основами программирования на языке Python. Я изучил среду разработки и освоил основные типы данных, такие как целые числа, числа с плавающей запятой, строки и логические значения. Также я разобрался с командами ввода и вывода данных, что позволяет мне взаимодействовать с программой и получать от неё результаты.

Особое внимание я уделил структуре ветвления, которая позволяет мне создавать программы, способные принимать решения в зависимости от условий. Я изучил конструкции `'if'`, `'if-else'` и `'if-elif-else'`, которые позволяют мне создавать логические условия и выполнять различные действия в зависимости от их истинности.

Также я познакомился с циклическими конструкциями `'for'` и `'while'`, которые позволяют мне выполнять определенный блок кода несколько раз. Циклы позволяют мне автоматизировать повторяющиеся действия и создавать более сложные алгоритмы.

Полученные знания являются основой для дальнейшего изучения программирования на Python. В будущем я смогу использовать эти знания для решения более сложных задач, создания собственных программ и разработки интересных проектов.

Ответы на вопросы:

1. Шесть основных причин, по которым программисты выбирают Python:
 - a. Лёгкий для изучения синтаксис.
 - b. Простота поддержки кода.
 - c. Динамическая типизация и интерпретируемый характер.
 - d. Гибкость для быстрой разработки и тестирования.
 - e. Богатая экосистема для автоматизации.
 - f. Бесшовная интеграция с другими технологиями.
2. Четыре известные компании или организации, использующие Python:
 - a. Яндекс,
 - b. Вконтакте/Mail.ru,
 - c. OpenAI,
 - d. Google.
3. Причины, по которым вы можете не захотеть использовать Python в

приложениях:

- a. Низкая производительность по сравнению с компилируемыми языками.
 - b. Отсутствие некоторых низкоуровневых возможностей, таких как прямой доступ к памяти.
4. Задачи, которые можно решать с помощью Python:
 - a. Научные вычисления,
 - b. Обработка данных,
 - c. Машинное обучение и искусственный интеллект,
 - d. Автоматизация процессов,
 - e. Веб-разработка,
 - f. Кибербезопасность.
5. Интерпретатор Python — это программа, которая выполняет исходный программный код, написанный на языке Python.
6. Исходный программный код — это текст программы, написанный на языке Python, который интерпретатор преобразует в машинный код и затем выполняет.
7. Чтобы запустить интерактивный сеанс работы с интерпретатором Python, нужно открыть командную строку и ввести команду `python` или `python3`.
8. Чтобы запустить сценарий из среды разработки IDLE, нужно открыть файл сценария в редакторе и нажать клавишу F5 или нажать в меню:
 - a. «Run» → «Run Module».
9. Результат выражения `1 + 2.0 + 3` будет иметь тип `float`, так как сумма состоит из чисел с плавающей точкой.
10. Три способа присвоить одно и то же значение трём переменным:
 - a. Использование цикла `for`,
 - b. Использование списка,
 - c. Использование кортежа.
11. Чтобы оформить инструкцию `if/else` в виде выражения, используется конструкция

```
if (усл.):  
    (действия)
```

`else:` (иначе эти действия)

12. Слова `True` и `False` обозначают логические значения «истина» и «ложь»

13. Блок `else` в циклах выполняется, когда условие цикла становится ложным.