

Итоговая проверочная работа.

Данная работа необходима для проверки ваших знаний и навыков по итогу прохождения первого блока обучения на программе разработчик. Мы должны убедиться что базовое знакомство с it прошло успешно.

Задача алгоритмически не самая сложная, однако для полноценного выполнения Проверочной работы Необходимо:

- 1. Создать репозиторий на Github
- 2. Нарисовать блок-схему алгоритма (можно обойтись блок-схемой основной содержательной части, если вы выделяете ее в отдельный метод)
- 3. Снабдить репозиторий оформленным текстовым описанием решения (файл README.md)
- 4. Написать программу, решающую поставленную задачу
- 5. Использовать контроль версий в работе над этим небольшим проектом (не должно быть так что все залито одним коммитом, как минимум этапы 2, 3 и 4 должны быть расположены в разных коммитах)

Задача: Написать программу, которая из имеющегося массива строк формирует массив из строк, длина которых меньше либо равна 3 символа. Первоначальный массив можно ввести с клавиатуры, либо задать на старте выполнения алгоритма. При решении не рекомендуется пользоваться коллекциями, лучше обойтись исключительно массивами.

Примеры:

```
["hello", "2", "world,":-)]->["2",":-)"]
```

```
["1234","1567","-2","computerScience"]->["-2"]
```

```
["Russia","Denmark","Kazan"]->[]
```

Я рад помочь в написании программы, которая решает поставленную задачу. Вот пример кода на Python:

```
# Ввод исходного массива с клавиатуры
arr = input("Введите массив строк, разделенных пробелами: ").split()

# Создание пустого массива для хранения результатов
result = []

# Проход по каждой строке исходного массива
for s in arr:
    # Добавление строки в результат, если ее длина меньше или равна 3 символам
    if len(s) <= 3:
        result.append(s)

# Вывод результата на экран
print(result)
```

```
# Input the initial array from the keyboard
arr = input("Enter an array of strings separated by spaces: ").split()

# Create an empty array to store the results
result = []

# Loop through each string in the initial array
for s in arr:
    # Add the string to the result if its length is less than or equal to 3
    if len(s) <= 3:
        result.append(s)

# Print the result to the screen
print(result)
```

Этот код сначала запрашивает у пользователя массив строк, разделенных пробелами, затем создает пустой массив для хранения результатов и проходит по каждой строке исходного массива. Если длина строки не превышает 3 символов, она добавляется в массив результатов. В конце программа выводит на экран массив результатов.

Для контроля версий этот код можно сохранить в файле `main.py` в отдельной папке, затем выполнить команды `git init`, `git add main.py`, `git commit -m "Initial commit"` для создания репозитория и первого коммита. Затем можно добавить файл `README.md` и выполнить команду `git commit -m "Add README.md"`. После этого можно добавить блок-схему алгоритма и выполнить команду `git commit -m "Add algorithm flowchart"`. И наконец, после написания оформленного текстового описания решения в файле `README.md`, можно выполнить команду `git commit -m "Add solution description"`.

Final Assessment Work.

This work is necessary to evaluate your knowledge and skills at the end of the first training block for the developer program. We need to make sure that the basic introduction to IT has been successful.

The algorithmic task itself is not the most difficult, however, for the full implementation of the Assessment Work, it is necessary to:

1. Create a repository on Github.

2. Draw a flowchart of the algorithm (you can use a flowchart of the main meaningful part if you highlight it in a separate method).

3. Provide the repository with a formatted text description of the solution (README.md file).

4. Write a program that solves the assigned task.

5. Use version control in working on this small project (it should not be that everything is uploaded in one commit, at least steps 2, 3, and 4 should be located in different commits).

Task: Write a program that forms an array of strings from the given array of strings whose length is less than or equal to 3. The initial array can be entered from the keyboard or set at the start of the algorithm execution. When solving the problem, it is not recommended to use collections, it is better to use arrays exclusively.

Examples:
["hello", "2", "world", ":-)"] -> ["2", ":-)"]
["1234", "1567", "-2", "computerScience"] -> ["-2"]
["Russia", "Denmark", "Kazan"] -> []

Итоговая проверочная работа

Данная работа необходима для проверки ваших знаний и навыков по итогу прохождения первого блока обучения на программе разработчик. Мы должны убедиться, что базовое знакомство с IT прошло успешно.

Задача

Написать программу, которая из имеющегося массива строк формирует массив из строк, длина которых меньше либо равна 3 символа. Первоначальный массив можно ввести с клавиатуры, либо задать на старте выполнения алгоритма. При решении не рекомендуется пользоваться коллекциями, лучше обойтись исключительно массивами.

Примеры

- ["hello", "2", "world", ":-)"] -> ["2", ":-)"]
- ["1234", "1567", "-2", "computerScience"] -> ["-2"]
- ["Russia", "Denmark", "Kazan"] -> []

Решение

Блок-схема алгоритма

```
+-----+
| Start |
+-----+
  |
  v
+-----+
| Create inputArray |
+-----+
  |
  v
+-----+
| Create outputArray |
+-----+
  |
  v
+-----+      +-----+
| For each element |---->| If length(element) <= 3 |
| in inputArray   |      +-----+
```



Здесь представлена блок-схема алгоритма, который позволяет решить задачу. Программа состоит из нескольких шагов, каждый из которых выполняет определенную операцию. Сначала создаются два массива, затем выполняется цикл по элементам входного массива, в ходе которого проверяется условие и добавляются элементы в выходной массив. В конце алгоритма полученный массив возвращается.

Описание решения

1. Создаем массив строк `inputArray`.
2. Создаем пустой массив строк `outputArray`.
3. Проходим циклом по каждому элементу массива `inputArray`.
4. Если длина текущей строки меньше или равна 3 символам, то добавляем ее в массив `outputArray`.
5. Возвращаем массив `outputArray`.

Использование

Для использования программы необходимо запустить файл `main.py`. При запуске программы можно ввести массив строк с клавиатуры или задать его в коде. Результатом работы программы будет массив строк, длина которых меньше или равна 3 символам.

Контроль версий

Для контроля версий был создан репозиторий на Github. Работа над проектом велась в отдельных коммитах, разделяющих этапы выполнения задачи. Каждый коммит содержит изменения, относящиеся к конкретному этапу работы.