МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение   
высшего образования

НИЖЕГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ

УНИВЕРСИТЕТ им. Р.Е.АЛЕКСЕЕВА

Институт Радиоэлектроники и Информационных Технологий

Отчёт

по лабораторной работе №2

«Реализация пошаговых блок-схем алгоритмов» по вариантам»

ПРЕПОДАВАТЕЛЬ:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Степаненко М.А.

(подпись) (фамилия, и.,о.)

СТУДЕНТ:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Сенков И.Д.

(подпись) (фамилия, и.,о.)

24-ИВТ-3

(шифр группы)

Работа защищена «\_\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_

С оценкой \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Нижний Новгород 2025

**Задание:** Удалить из данной строки все повторения символов.

Описание алгоритма:

1. Начало алгоритма.

2. Вводим исходную строку (str).

3. Создаем пустую строку для хранения результата (result = "").

4. Инициализируем пустое множество для отслеживания уже встреченных символов (seen = пустое множество).

5. Устанавливаем индекс i равным 0.

6. Проверяем, меньше ли i длины строки str. Если да, продолжаем, если нет, переходим к шагу 11.

7. Получаем текущее значение символа char из исходной строки по индексу i

8. Проверяем, содержится ли char в множестве seen:

- Если нет, добавляем char в строку результата и в множество seen.

- Если да, просто пропускаем этот символ.

9. Увеличиваем значение i на 1

10. Возвращаемся к шагу 6

11. Выводим финальную строку result, которая теперь содержит только уникальные символы

12. Завершаем алгоритм.

Рассмотрим выполнение алгоритма удаления повторяющихся символов на примере строки "hello world" шаг за шагом:

1. Инициализация:

- Создаем пустую строку result для хранения результата: result = ""

- Создаем пустой набор seen для отслеживания встреченных символов: seen = {}

2. Проход по строке:

- Начинаем цикл по каждому символу в строке "hello world".

3. Первый символ 'h':

- Проверяем, есть ли 'h' в seen: нет.

- Добавляем 'h' в result: result = "h"

- Добавляем 'h' в seen: seen = {'h'}

4. Второй символ 'e':

- Проверяем 'e' в seen: нет.

- Добавляем 'e' в result: result = "he"

- Добавляем 'e' в seen: seen = {'h', 'e'}

5. Третий символ 'l':

- Проверяем 'l' в seen: нет.

- Добавляем 'l' в result: result = "hel"

- Добавляем 'l' в seen: seen = {'h', 'e', 'l'}

6. Четвертый символ 'l':

- Проверяем 'l' в seen: да (уже встречен).

- Пропускаем этот символ.

7. Пятый символ 'o':

- Проверяем 'o' в seen: нет.

- Добавляем 'o' в result: result = "helo"

- Добавляем 'o' в seen: seen = {'h', 'e', 'l', 'o'}

8. Шестой символ ' ' (пробел):

- Проверяем пробел в seen: нет.

- Добавляем пробел в result: result = "helo "

- Добавляем пробел в seen: seen = {'h', 'e', 'l', 'o', ' '}

9. Седьмой символ 'w':

- Проверяем 'w' в seen: нет.

- Добавляем 'w' в result: result = "helo w"

- Добавляем 'w' в seen: seen = {'h', 'e', 'l', 'o', ' ', 'w'}

10. Восьмой символ 'o':

- Проверяем 'o' в seen: да (уже встречен).

- Пропускаем этот символ.

11. Девятый символ 'r':

- Проверяем 'r' в seen: нет.

- Добавляем 'r' в result: result = "helo wr"

- Добавляем 'r' в seen: seen = {'h', 'e', 'l', 'o', ' ', 'w', 'r'}

12. Десятый символ 'l':

- Проверяем 'l' в seen: да (уже встречен).

- Пропускаем этот символ.

13. Одиннадцатый символ 'd':

- Проверяем 'd' в seen: нет.

- Добавляем 'd' в result: result = "helo wrd"

- Добавляем 'd' в seen: seen = {'h', 'e', 'l', 'o', ' ', 'w', 'r', 'd'}

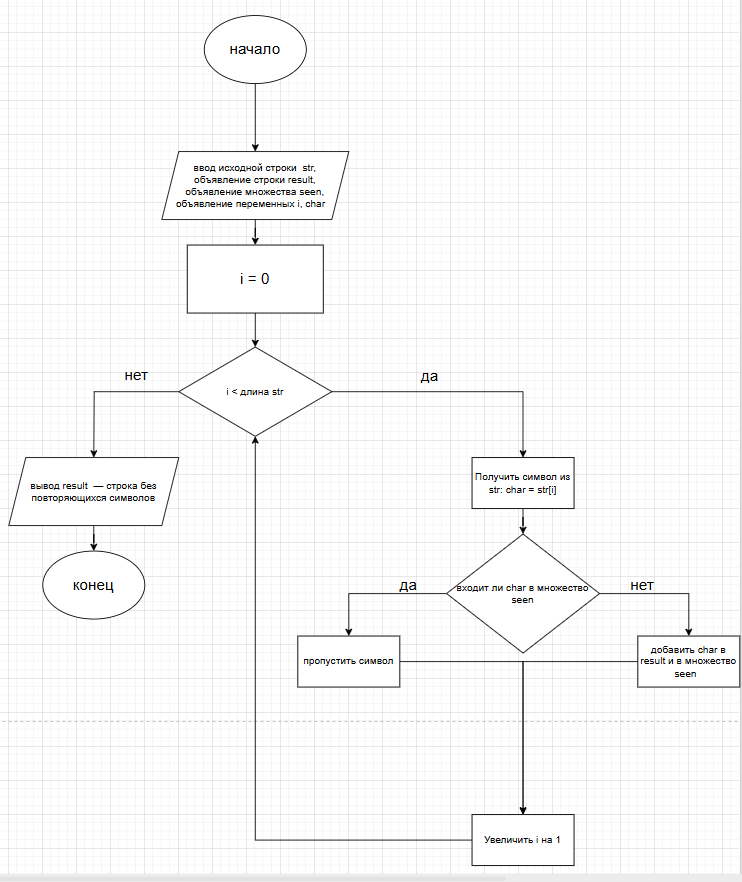
14. Завершение:

- Прошли по всем символам.

- Результат: "helo wrd".

Таким образом, итоговая строка без повторяющихся символов - "helo wrd".

Блок схема:



Пример работы алгоритма:

