

ГУАП

КАФЕДРА № 42

ОТЧЕТ
ЗАЩИЩЕН С ОЦЕНКОЙ
ПРЕПОДАВАТЕЛЬ

доцент

должность, уч. степень, звание

подпись, дата

В. А. Кузнецов

инициалы, фамилия

ОТЧЕТ О ЛАБОРАТОРНОЙ РАБОТЕ № 2.1

ПРОСТОЙ АЛГОРИТМ ОБРАБОТКИ СТРОК

по курсу:

ОСНОВЫ ПРОГРАММИРОВАНИЯ

РАБОТУ ВЫПОЛНИЛ

СТУДЕНТ гр. №

4326

подпись, дата

Г. С. Томчук

инициалы, фамилия

Санкт-Петербург 2024

СОДЕРЖАНИЕ

1 Постановка задачи.....	3
2 Схема алгоритма решения.....	4
3 Полное описание реализованной функции.....	5
3.1 extract_words_with_digits.....	5
3.2 contains_digit.....	5
3.3 free_words.....	6
3.4 main.....	6
4 Листинг программы.....	7
5 Несколько тестов работы программы.....	8

1 Постановка задачи

Задача: реализовать программную функцию на языке C/C++, выполняющую поставленную задачу. Вариант задания, пример входных и выходных данных представлен в таблице 1. Глобальные параметры использовать запрещено; допустимо использование дополнительных функций. Обработку строк необходимо осуществлять как массив символов, без использования специальных библиотек для работы со строками.

Таблица 1 – Вариант

N	Текст задания	Вход	Выход
8	Дана строка. Слово есть последовательность любых символов, разделенных пробелом. Сохранить отдельно все слова, содержащие цифры.	"K2y7 Smt g6 PH 4a"	"K2y7", "g6", "4a"

2 Схема алгоритма решения

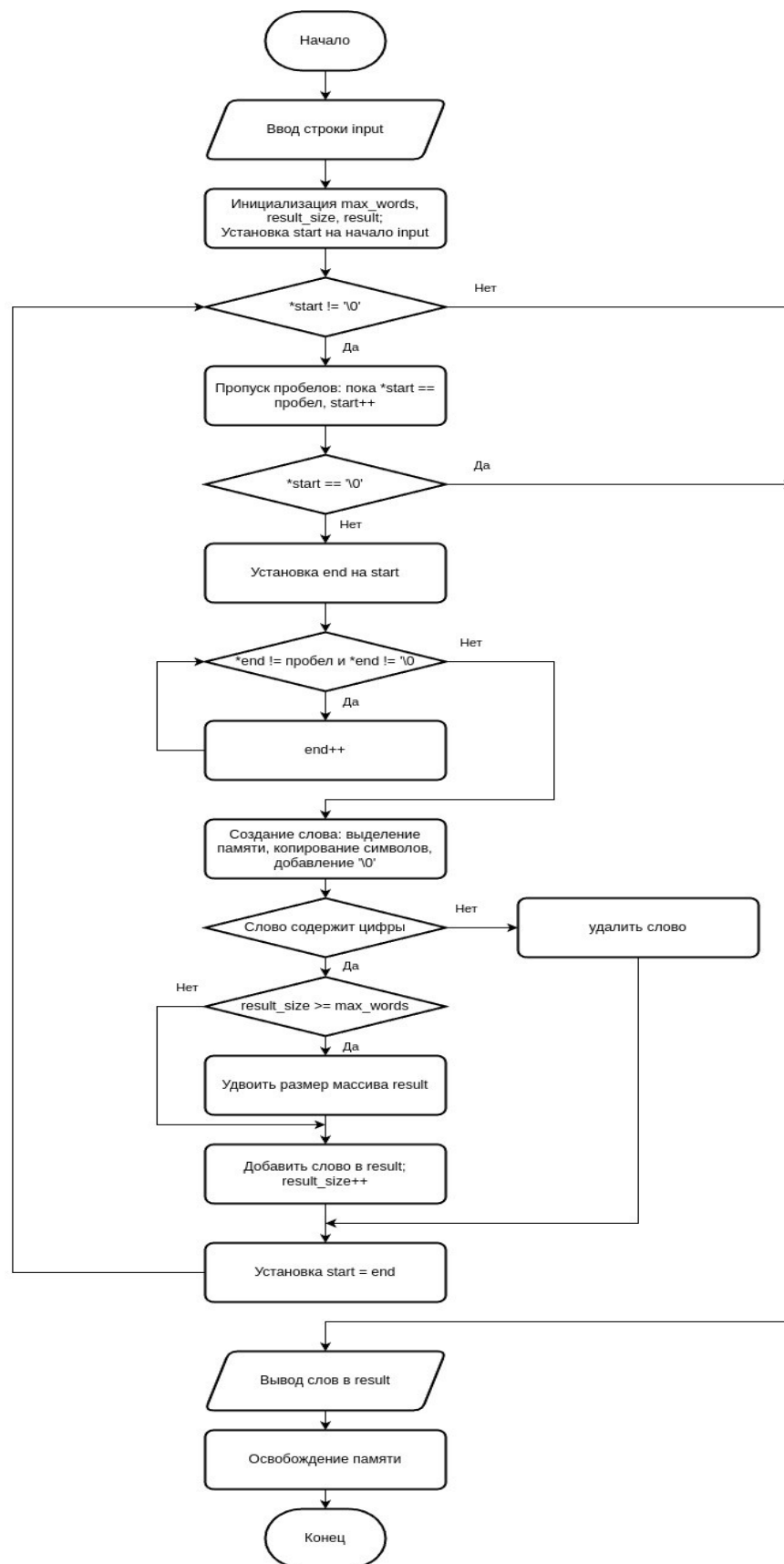


Рисунок 1 – Блок-схема алгоритма

3 Полное описание реализованной функции

3.1 `extract_words_with_digits`

Функция `extract_words_with_digits` извлекает из строки все слова, содержащие цифры, и сохраняет их в массиве. Принимает следующие аргументы:

1. `const char *input`: указатель на исходную строку.
2. `char **&result`: ссылка на указатель массива строк, в который будут сохранены слова, содержащие цифры.
3. `size_t &result_size`: ссылка на переменную, в которую будет записано количество найденных слов.

Работа функции происходит следующим образом:

1. Инициализация переменных: начальный размер массива слов (`max_words`), указатель на массив строк (`result`), и размер массива (`result_size`).
2. Указатель `start` устанавливается на начало входной строки.
3. Цикл проходит по всей строке, пропуская начальные пробелы.
4. Нахождение конца текущего слова (до первого пробела).
5. Копирование слова в новый массив символов.
6. Проверка, содержит ли слово цифры с помощью `contains_digit`.
7. Если слово содержит цифры, добавляем его в массив `result`.
8. При необходимости увеличиваем размер массива `result`.
9. Переход к следующему слову.

3.2 `contains_digit`

Функция `contains_digit` проверяет, содержит ли переданное слово хотя бы одну цифру. Принимает следующие аргументы:

1. `const char *word`: указатель на строку (слово), которую нужно проверить.

Работа функции происходит следующим образом:

1. Инициализация указателя на начало строки.
2. Цикл проходит по каждому символу строки.
3. Проверка, является ли текущий символ цифрой.

4. Если символ - цифра, возвращает true.
5. Если цикл завершился без нахождения цифры, возвращает false.

3.3 free_words

Функция free_words освобождает память, выделенную под массив строк.

Принимает следующие аргументы:

1. char **words: указатель на массив строк.
2. size_t size: количество строк в массиве.

Работа функции происходит следующим образом:

1. Цикл проходит по каждому элементу массива words.
2. Освобождает память, выделенную под каждое слово.
3. Освобождает память, выделенную под массив указателей.

3.4 main

1. Объявление массива input для ввода строки.
2. Ввод строки от пользователя.
3. Объявление указателя words_with_digits и переменной words_with_digits_size.
4. Вызов extract_words_with_digits для извлечения слов, содержащих цифры.
5. Вывод слов, содержащих цифры, на экран.
6. Освобождение памяти, выделенной под массив слов с цифрами, с помощью free_words.
7. Возвращение 0 для обозначения успешного завершения программы.

4 Листинг программы

Листинг 1

```
#include <iostream>

bool contains_digit(const char *word) {
    while (*word) {
        if (std::isdigit(*word))
            return true;
        ++word;
    }
    return false;
}

void extract_words_with_digits(const char *input, char **&result, size_t
&result_size) {
    size_t max_words = 8; // Начальный размер массива слов
    result = new char *[max_words];
    result_size = 0;

    const char *start = input;
    const char *end;

    while (*start) {
        // Пропуск пробелов в начале
        while (*start && std::isspace(*start))
            ++start;

        if (!*start) break;

        // Находим конец слова
        end = start;
        while (*end && !std::isspace(*end)) {
            ++end;
        }

        // Копирование слова в отдельный массив
        long length = end - start;
        char *word = new char[length + 1];
        for (size_t i = 0; i < length; ++i)
            word[i] = start[i];
        word[length] = '\0';

        // Проверка, содержит ли слово цифры
        if (contains_digit(word)) {
            if (result_size ≥ max_words) {
                // Увеличиваем размер массива
                max_words *= 2;
                char **new_result = new char *[max_words];
                for (size_t i = 0; i < result_size; ++i)
                    new_result[i] = result[i];
                delete[] result;
                result = new_result;
            }
            result[result_size++] = word;
        } else {
            delete[] word;
        }
    }
}
```

```

    }

    start = end;
}
}

void free_words(char **words, size_t size) {
    for (size_t i = 0; i < size; ++i)
        delete[] words[i];
    delete[] words;
}

int main() {
    char input[256];
    std::cout << "Введите строку: ";
    std::cin.getline(input, 256);

    char **words_with_digits;
    size_t words_with_digits_size;

    extract_words_with_digits(input, words_with_digits,
words_with_digits_size);

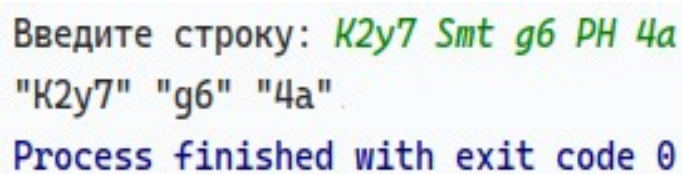
    for (size_t i = 0; i < words_with_digits_size; ++i)
        std::cout << "\"" << words_with_digits[i] << "\" ";

    // Освобождаем память
    free_words(words_with_digits, words_with_digits_size);

    return 0;
}

```

5 Несколько тестов работы программы

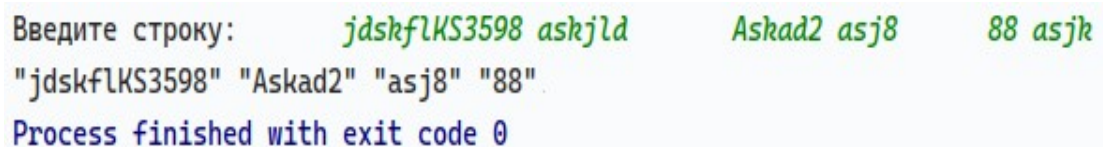


```

Введите строку: K2y7 Smt g6 PH 4a
"K2y7" "g6" "4a"
Process finished with exit code 0

```

Рисунок 2



```

Введите строку: jdskfLKS3598 askjld Askad2 asj8 88 asjk
"jdskfLKS3598" "Askad2" "asj8" "88"
Process finished with exit code 0

```

Рисунок 3


```
Введите строку: абвгд123 фылдов длы5ло 000 лыв  
"абвгд123" "длы5ло" "000"  
Process finished with exit code 0
```

Рисунок 4