**Лабораторная работа №3. Обработка потока символов**

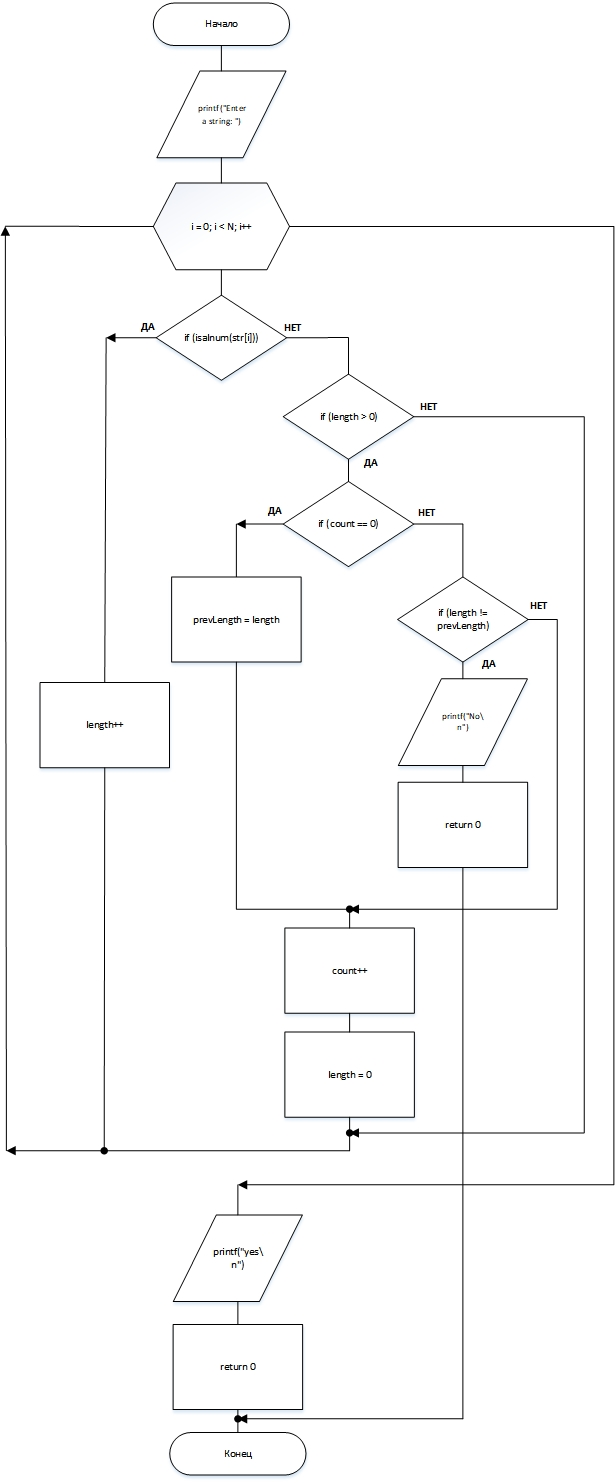
Цель работы: изучение организации ввода/вывода потока символов и разбиения его на лексемы.

Задача 3.12

Определить, являются ли все слова из потока символов словами одинаковой

длины.

Блок-схема

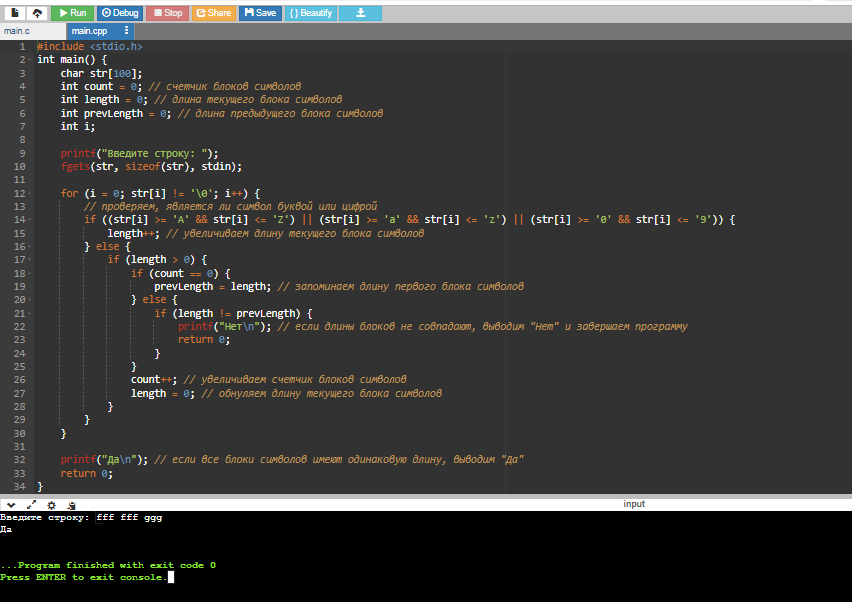


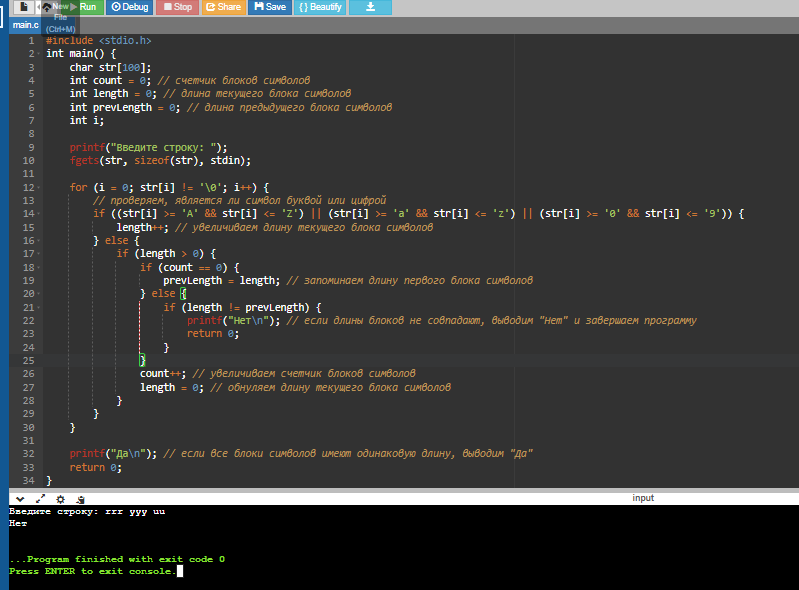
Исходный код программы

#include <stdio.h>   
  
int main() {   
    // Объявление массива символов для хранения введенной строки  
    char str[100];   
    // Счетчик для подсчета количества серий цифр и букв  
    int count = 0;   
    // Переменная для хранения длины текущей серии  
    int length = 0;   
    // Переменная для хранения длины предыдущей серии  
    int prevLength = 0;   
    // Индекс для обхода массива  
    int i = 0;   
    // Переменная для хранения текущего введенного символа  
    int c;   
  
    // Вывод запроса на ввод строки  
    printf("Enter a string: ");   
  
    // Чтение строки до символа новой строки или конца файла  
    while ((c = getchar()) != '\n' && c != EOF) {   
        str[i++] = c;   
    }   
    // Добавление завершающего нулевого символа в строку  
    str[i] = '\0';   
  
    // Обход строки для анализа символов  
    for (i = 0; str[i] != '\0'; i++) {   
        // Проверка, является ли символ буквой или цифрой  
        if ((str[i] >= 'A' && str[i] <= 'Z') || (str[i] >= 'a' && str[i] <= 'z')  || (str[i] >= '0' && str[i] <= '9')) {   
            length++;   
        } else {   
            // Если серия не пустая  
            if (length > 0) {   
                // Первая серия  
                if (count == 0) {   
                    prevLength = length;   
                } else {   
                    // Проверка на совпадение длины текущей серии с предыдущей  
                    if (length != prevLength) {   
                        printf("No\n");   
                        return 0;   
                    }   
                }   
                count++;   
                length = 0;   
            }   
        }   
    }   
  
    // Проверка последней серии, если она не завершена символом новой строки  
    if (length > 0 && length != prevLength) {  
        printf("No\n");  
        return 0;  
    }

printf("Yes\n");

return 0;  
}





Доп задание:

Теория:

1.Стек вызова функции (call stack) - это структура данных, используемая для отслеживания вызовов функций в программе. В контексте вашего кода, стек вызова функции будет содержать информацию о вызове функции main().

2.В языке Си для определения текущего состояния переменной (то есть ее значения) Вы можете использовать оператор `printf()` , чтобы вывести значение переменной на экран.

Практика:

#include <stdio.h>

#include <signal.h>

int main() {

char str[100];

int count = 0; // счетчик блоков символов

int length = 0; // длина текущего блока символов

int prevLength = 0; // длина предыдущего блока символов

int i;

printf("Введите строку: ");

fgets(str, sizeof(str), stdin);

for (i = 0; str[i] != '\0'; i++) {

// проверяем, является ли символ буквой или цифрой

if ((str[i] >= 'A' && str[i] <= 'Z') || (str[i] >= 'a' && str[i] <= 'z') || (str[i] >= '0' && str[i] <= '9')) {

length++; // увеличиваем длину текущего блока символов

} else {

if (length > 0) {

if (count == 0) {

prevLength = length; // запоминаем длину первого блока символов

} else {

if (length != prevLength) {

printf("Нет\n"); // если длины блоков не совпадают, выводим "Нет" и завершаем программу

return 0;

}

}

count++; // увеличиваем счетчик блоков символов

length = 0; // обнуляем длину текущего блока символов

}

}

}

raise(SIGINT);

printf("Да\n"); // если все блоки символов имеют одинаковую длину, выводим "Да"

return 0;

}