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

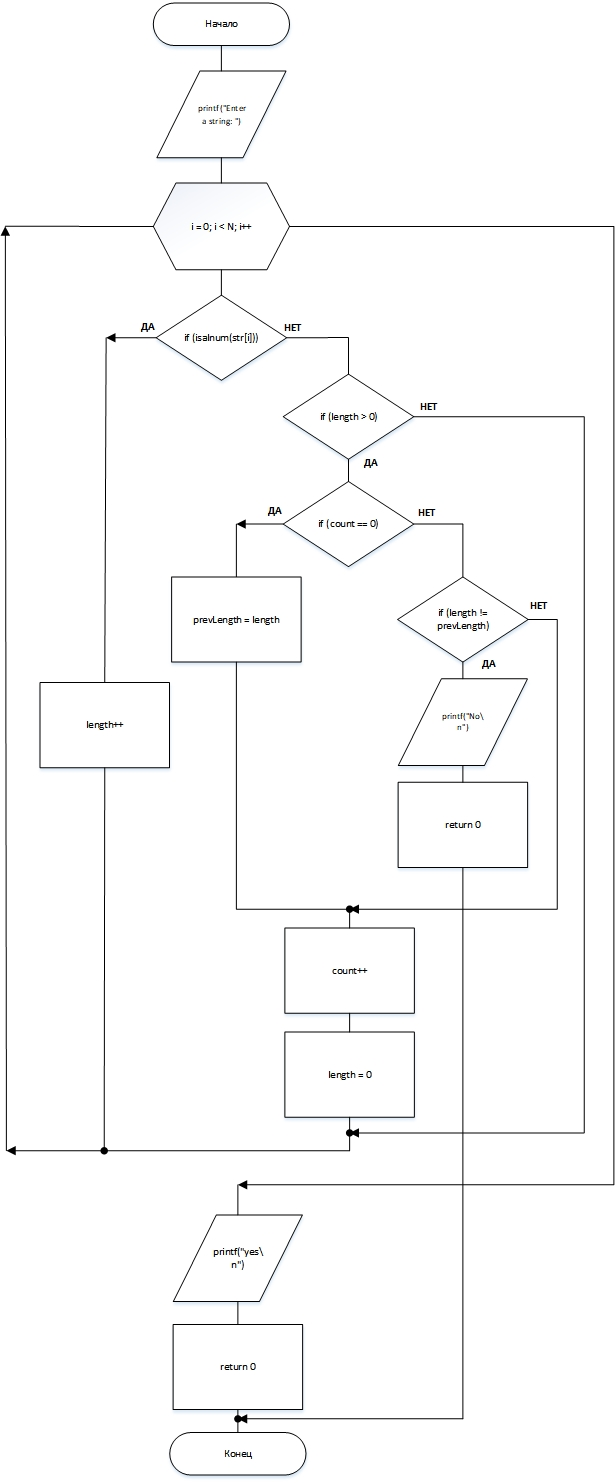
Цель работы: изучение организации ввода/вывода потока символов и разбиения его на лексемы.

Задача 3.12

Определить, являются ли все слова из потока символов словами одинаковой

длины.

Блок-схема



Исходный код программы

#include <stdio.h>

int main() {

char str[100]; // Объявление массива символов для хранения введенной строки

int count = 0; // Счетчик для подсчета количества серий цифр и букв

int length = 0; // Переменная для хранения длины текущей серии

int prevLength = 0; // Переменная для хранения длины предыдущей серии

int i = 0; // Индекс для обхода массива

int c; // Переменная для хранения текущего введенного символа

printf("Enter a string: "); // Вывод запроса на ввод строки

while ((c = getchar()) != '\n' && c != EOF) { // Чтение строки до символа новой строки или конца файла

str[i++] = c;

}

str[i] = '\0'; // Добавление завершающего нулевого символа в строку

for (i = 0; str[i] != '\0'; i++) { // Обход строки для анализа символов

if ((str[i] >= 'A' && str[i] <= 'Z') (str[i] >= 'a' && str[i] <= 'z') (str[i] >= '0' && str[i] <= '9')) { // Проверка, является ли символ буквой или цифрой

length++;

} else {

if (length > 0) { // Если серия не пустая

if (count == 0) { // Первая серия

prevLength = length;

} else {

if (length != prevLength) { // Проверка на совпадение длины текущей серии с предыдущей

printf("No\n");

return 0;

}

}

count++;

length = 0;

}

}

}

if (length > 0 && length != prevLength) { // Проверка последней серии, если она не завершена символом новой строки

printf("No\n");

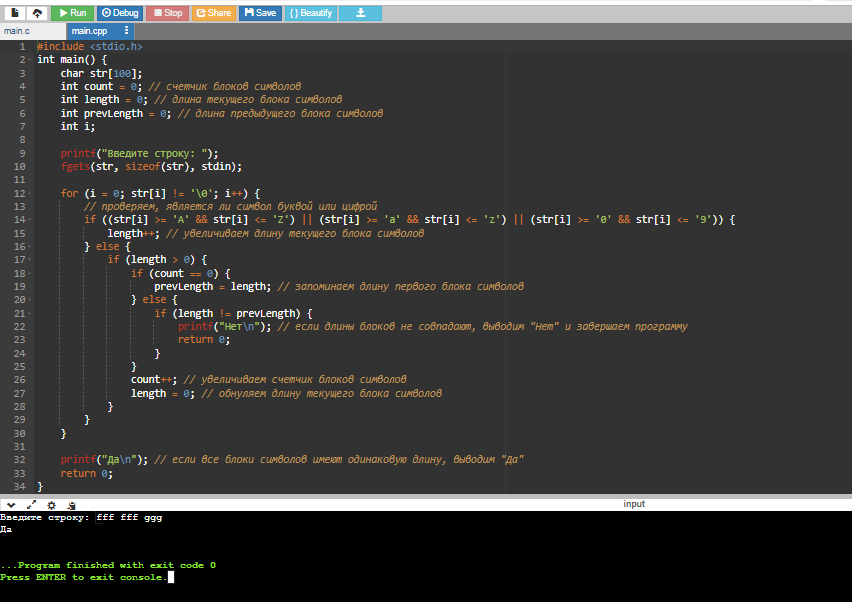
return 0;

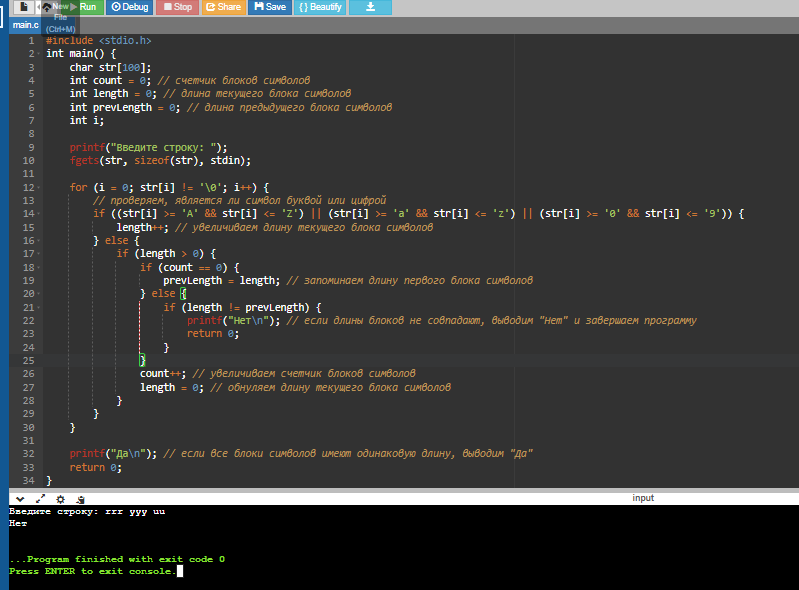
}

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;

}