Министерство образования Республики Беларусь

Учреждение образования

БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ

Факультет компьютерных систем и сетей

Кафедра электронных вычислительных машин

ОТЧЁТ

к лабораторной работе №7

на тему

РАБОТА СО СТРОКАМИ

ВАРИАНТ 6

Проверил Выполнил

асс. каф. ЭВМ студент гр. 318302 П.С. Жуковец В.С.Бахур

МИНСК 2023

**1** **ЗАДАНИЕ №1**

**1.1 Условие**

1. Дана строка. Найти количество повторений введенного слова
2. Даны строки S и S0. Проверить, содержится ли строка S0 в строке S. Если содержится, то вывести «True», если не содержится, то вывести «False».

**1.2 Алгоритм решения**

Алгоритм предоставлен в «Приложении A»; Рисунок 1.1 – 1.2

**1.3 Листинг программы**

/\*1. Дана строка. Найти количество повторений введенного слова

2. Даны строки S и S0. Проверить, содержится ли строка S0 в строке S. Если содержится, то вывести «True», если не содержится, то вывести «False».\*/

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <stdio.h>

#include <stdlib.h>

// Функция для ввода строки с динамическим выделением памяти

char\* inputString() {

char\* str; // Указатель на строку, которую мы будем вводить

int ch, size = 10, len = 0; // Переменные для символа, размера строки и текущей длины строки

// Выделяем начальную память под строку

str = realloc(NULL, sizeof(char) \* size);

// Проверка успешного выделения памяти

if (!str) return str;

// Чтение символов с клавиатуры, пока не достигнут конец файла (EOF) или символ новой строки

while (EOF != (ch = fgetc(stdin)) && ch != '\n') {

str[len++] = ch;

// Если достигнут лимит размера, увеличиваем размер строки

if (len == size) {

str = realloc(str, sizeof(char) \* (size += 16));

// Проверка успешного выделения памяти после увеличения

if (!str) return str;

}

}

// Добавляем завершающий символ нуля к строке

str[len++] = '\0';

// Перераспределяем память, чтобы убрать лишний ненужный запас

return realloc(str, sizeof(char) \* len);

}

int isWordBoundary(char c) {

return c == ' ' || c == '\t' || c == '\n' || c == '\0';

}

int countOccurrences(char\* line, char\* word) {

int count = 0, wordLen = 0, lineLen = 0, i = 0, j = 0;

while (word[wordLen] != '\0') wordLen++;

while (line[lineLen] != '\0') lineLen++;

for (i = 0; i <= lineLen - wordLen; i++) {

if ((i == 0 || isWordBoundary(line[i - 1])) && isWordBoundary(line[i + wordLen])) {

for (j = 0; j < wordLen; j++) {

if (line[i + j] != word[j])

break;

}

if (j == wordLen) {

count++;

i += wordLen - 1;

}

}

}

return count;

}

// Функция проверки, является ли строка S подстрокой строки S0

int isSubstring(char\* S, char\* S0) {

// Переменные для хранения длин строк S и S0, и индексы для циклов

int SLen = 0, S0Len = 0, i = 0, j = 0;

// Определение длины строки S

while (S[SLen] != '\0') {

SLen++;

}

// Определение длины строки S0

while (S0[S0Len] != '\0') {

S0Len++;

}

// Итерации по строке S для поиска подстроки S0

for (i = 0; i <= SLen - S0Len; i++) {

// Итерации по символам подстроки S0

for (j = 0; j < S0Len; j++) {

// Сравнение символов текущих позиций в строках S и S0

if (S[i + j] != S0[j])

break; // Прерывание цикла, если символы не совпадают,

}

// Если внутренний цикл завершился полностью, значит, найдена подстрока S0 в строке S

if (j == S0Len)

return 1; // Возвращаем 1, если подстрока найдена

}

// Если цикл не завершился досрочно, подстрока не найдена, возвращаем 0

return 0;

}

void task1() {

printf("Введите строку: ");

char\* line = inputString();

printf("Введите слово: ");

char\* word = inputString();

printf("Слово '%s' встречается %d раз.\n", word, countOccurrences(line, word));

free(line);

free(word);

}

void task2() {

printf("Введите строку S: ");

char\* S = inputString();

printf("Введите строку S0: ");

char\* S0 = inputString();

if (isSubstring(S, S0)) {

printf("true\n");

}

else {

printf("false\n");

}

free(S);

free(S0);

}

int main(void) {

system("chcp 1251 > nul");

int choice;

void(\*task)() = NULL;

Start:

printf("Введите номер задачи (1 или 2): ");

scanf("%d", &choice);

getchar(); // Удаляем символ новой строки после ввода числа

if (choice == 1) {

task = task1;

task();

}

else if (choice == 2) {

task = task2;

task();

}

else {

printf("Неверный номер задачи.\n");

goto Start;

}

return 0;

}

**Приложение А**

Блок-схема алгоритма 1

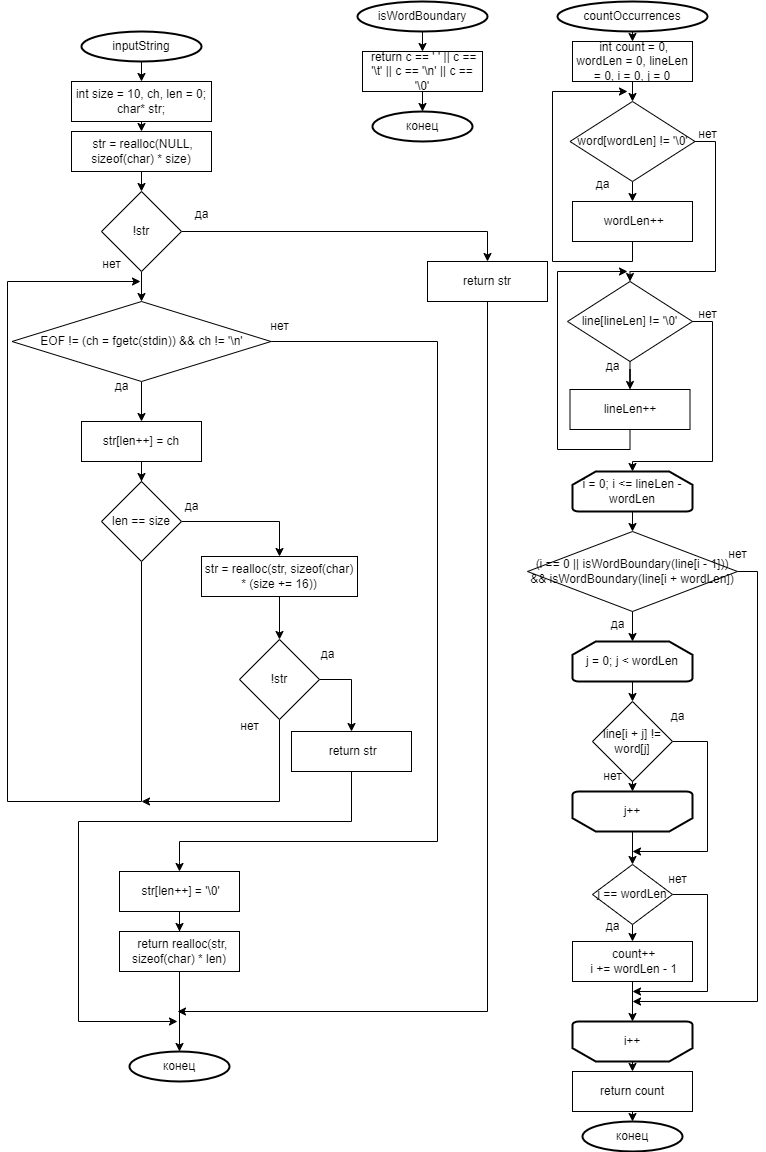


Рисунок 1.1 – Блок-схема №1

Блок-схема алгоритма 1

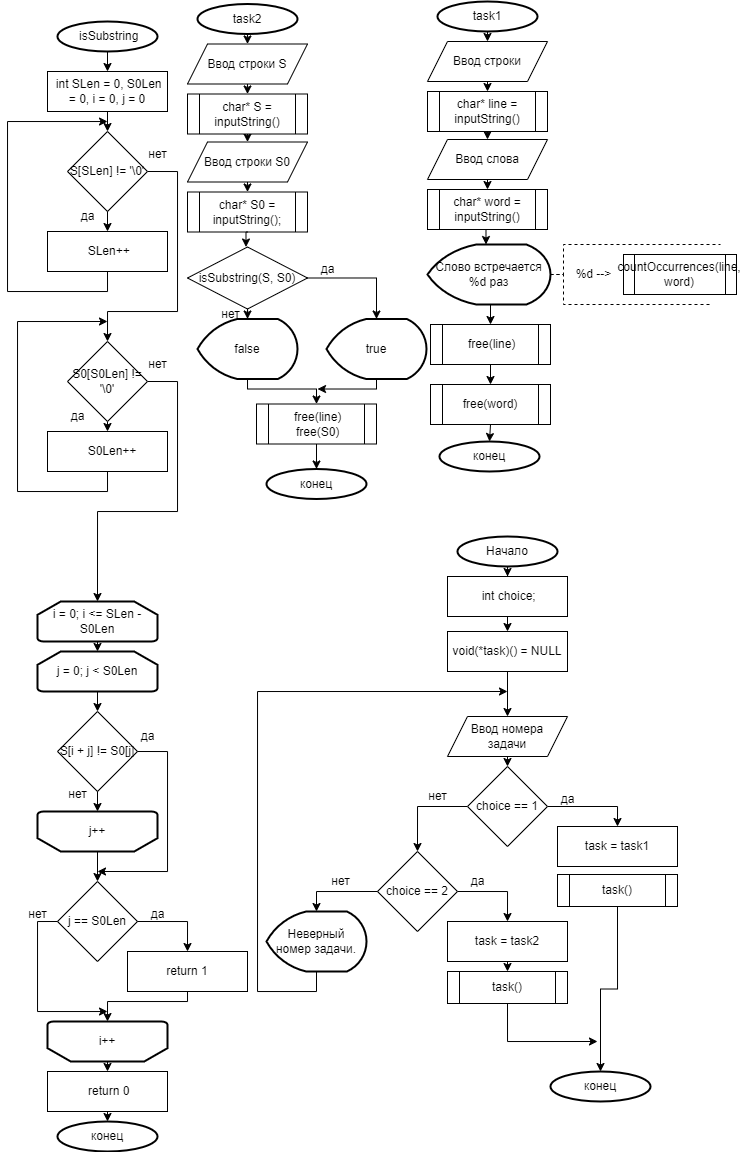


Рисунок 1.2 – Блок-схема №1