Министерство образования Республики Беларусь

Учреждение образования

БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ

Факультет компьютерных систем и сетей

Кафедра программного обеспечения информационных технологий

Компиляторные технологии

ОТЧЕТ

по лабораторной работе

на тему:

ПРЕОБРАЗОВАНИЕ ТЕКСТОВ С ИСПОЛЬЗОВАНИЕМ ГЕНЕРАТОРА ЛЕКСИЧЕСКИХ АНАЛИЗАТОРОВ FLEX

|  |  |
| --- | --- |
| Выполнил  Студент гр. 351003 | Е. Д. Гузаев |
| Проверил | С. В. Болтак |

Минск, 2024

**Вариант 11 ((15 – 4) mod 15)**

**Задание:**

|  |  |  |
| --- | --- | --- |
| 11 | Текстовый документ | Пронумеровать строки. Результатом работы должен быть текстовый документ, в котором перед началом каждой строки записан её порядковый номер. |

Программная реализация

#include <stdio.h>

#include <string.h>

// тип перечисления включает классы символов

enum charType {

ctUnknown,

ctDecDigit,

ctHexDigit,

ctLetter

};

// двумерный массив автоматических переходов

const int Transitions[4][4] = {

{0, 0, 0, 0},

{0, 2, 0, 0},

{0, 2, 2, 3},

{0, 0, 0, 0}

};

const int isFinalState[4] = {

0, 0, 0, 1

};

// функция для получения класса символа

int getCharType(char chr) {

if (chr >= '0' && chr <= '9')

return ctDecDigit;

if (chr >= 'A' && chr <= 'F')

return ctHexDigit;

if (chr == 'b' || chr == 'O' || chr == 'o' || chr == 'h' || chr == 'H')

return ctLetter;

return ctUnknown;

}

// функция для сброса автомата

void reset(int \*state, int \*startPos, int \*pos, int mas[]) {

\*state = 1;

\*startPos = \*pos + 1;

mas[0] = 1;

mas[1] = 1;

}

// главная функция для поиска чисел

void searchString(char str[]) {

int state = 1;

int i = 0;

int startPos = 0;

int isCorrect[] = {1, 1};

int count = 0;

int j;

printf("Найденные числа: \n");

while (i < strlen(str)) {

state = Transitions[state][getCharType(str[i])];

if (isFinalState[state] != 1 && str[i] != '0' && str[i] != '1')

isCorrect[0] = 0;

if (isFinalState[state] != 1 && (getCharType(str[i]) != 1 || str[i] > '7'))

isCorrect[1] = 0;

if (isFinalState[state] == 1) {

if (str[i] == 'b' && isCorrect[0] == 0) {

j = i - 1;

while (str[j] == '0' || str[j] == '1')

j--;

j++;

if (j != i) {

while (j <= i) {

putchar(str[j]);

j++;

}

count++;

printf("\n");

}

reset(&state, &startPos, &i, isCorrect);

} else {

if ((str[i] == 'o' || str[i] == 'O') && isCorrect[1] == 0) {

j = i - 1;

while (str[j] >= '0' && str[j] <= '7')

j--;

j++;

if (j != i) {

while (j <= i) {

putchar(str[j]);

j++;

}

count++;

printf("\n");

}

reset(&state, &startPos, &i, isCorrect);

} else {

count++;

for (j = startPos; j <= i; j++)

putchar(str[j]);

printf("\n");

reset(&state, &startPos, &i, isCorrect);

}

}

}

if (state == 0) {

reset(&state, &startPos, &i, isCorrect);

}

i++;

}

printf("Найдено чисел: %d\n", count);

}

// вспомогательная функция для проверки корректности подстроки

int isValidString(const char \*substr) {

int len = strlen(substr);

if (len < 2) {

return 0; // Минимальная длина должна быть 2: цифра + буква

}

// Первый символ должен быть десятичной цифрой

if (!(substr[0] >= '0' && substr[0] <= '9')) {

return 0;

}

// Промежуточные символы могут быть десятичными или шестнадцатеричными цифрами

for (int i = 1; i < len - 1; i++) {

if (!((substr[i] >= '0' && substr[i] <= '9') || (substr[i] >= 'A' && substr[i] <= 'F'))) {

return 0;

}

}

// Последний символ должен быть одной из разрешенных букв

char lastChar = substr[len - 1];

if (!(lastChar == 'b' || lastChar == 'O' || lastChar == 'o' || lastChar == 'H' || lastChar == 'h')) {

return 0;

}

return 1;

}

// функция для поиска всех подстрок и вывода корректных

void printValidSubstrings(char \*s) {

printf("Корректные подстроки: \n");

for (int i = 0; i < strlen(s); i++) {

for (int j = i + 1; j <= strlen(s); j++) {

char substr[100];

strncpy(substr, s + i, j - i);

substr[j - i] = '\0';

if (isValidString(substr))

printf("%s\n", substr);

}

}

}

// основная функция

int main() {

FILE \*out;

char input[100];

char choice;

printf("Введите 0, 1 или 2: 0 - ввод из файла, 1 - ввод с клавиатуры, 2 - поиск подстрок\n");

do {

choice = getchar();

getchar(); // чтобы захватить символ новой строки после getchar()

if (choice == '1') {

printf("Введите строку:\n");

printf("> ");

scanf("%s", input);

searchString(input);

}

if (choice == '0') {

out = fopen("C:\\Example.txt", "r");

fscanf(out, "%s", input);

printf("Строка из файла: %s\n", input);

searchString(input);

fclose(out);

}

if (choice == '2') {

printf("Введите строку:\n");

printf("> ");

scanf("%s", input);

printValidSubstrings(input);

}

if (choice != '1' && choice != '0' && choice != '2')

printf("Ошибка. Введите число заново.\n");

} while (choice != '0' && choice != '1' && choice != '2');

return 0;

}

**Тесты**

Тест 1

Тестовая ситуация: отсутствие правильных чисел

Исходное выражение = “abc”

Ожидаемый результат: Найдено чисел: 0;

Полученный результат: Найдено чисел: 0;

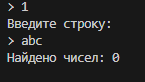


Рисунок 3 – Полученный результат. Тест 1

Тест 2

Тестовая ситуация: присутствие одного числа

Исходное выражение = “qew34H”

Ожидаемый результат: Найдено чисел: 1;

Полученный результат: Найдено чисел: 1;

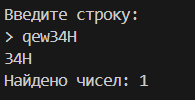


Рисунок 4 – Полученный результат. Тест 2

Тест 3

Тестовая ситуация: присутствие двух чисел в разных системах

счисления

Исходное выражение = “10101101010b23O”

Ожидаемый результат: Найдено чисел: 2;

Полученный результат: Найдено чисел: 2;

Полученный результат:

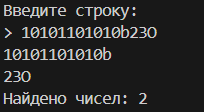


Рисунок 5 – Полученный результат. Тест 3

Тест 4

Тестовая ситуация: ввод строки с файла

Исходное выражение = “12231H”

Ожидаемый результат: Найдено чисел: 1;

Полученный результат: Найдено чисел: 1;

Полученный результат:

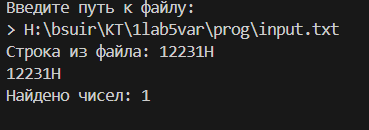


Рисунок 6 – Полученный результат. Тест 6