

1. **Цель работы**

Изучение организации ввода/вывода потока символов и разбиения его на лексемы.

1. **Постановка задачи**

Вариант 1: В потоке символов сосчитать число слов, у которых первые две буквы совпадают.

1. **Формализация задачи**

Для нахождения числа слов, у которых первые две буквы совпадают, необходимо разделять символы на слова и сравнивать первые два символа слова. Количество слов не может быть отрицательным. В число символов-разделителей входят: ' ', '.', '\n' и ','. Для решения задачи создаётся цикл проверки символов на то, являются ли они разделителями. Перед проверкой все символы приводятся к верхнему регистру. Символы между разделителями – слова. Первый символ слова изменяет флаг first и сохраняемся как первая буква. Вторая буква сравнивается с первой и в случае совпадения увеличивается счётчик.

1. **Набор тестовых примеров**

|  |  |
| --- | --- |
| Вводимые слова | Вывод программы |
|  | The amount of words: 0 |
| OOoOooo oOoOoo ooOO OOooOOO | The amount of words: 3 |
| Some rrandom texxttt hhhereee | The amount of words: 2 |

1. **Схема алгоритма**

Схема алгоритма программы представлена на рисунке 1.

Начало

c; first=1; letter1=0;

counter=0; endFlag=1.

Ввод c

c != EOF

c == ' ' || c == '.' || c == '\n' || c == ','

Нет

Нет

Да

Да

A

B

C

D

Нет

Нет

first == 1

D

C

B

A

first = 1;

Да

letter1==c

Да

1 - Count again.

2 - Exit.

1

Вывод результата counter

letter1 = 0;

first = 0;

letter1 = c;

2

counter++;

letter1 = 0;

Конец

*Рисунок 1 – Схема алгоритма программы.*

1. **Листинг программы**

#if defined(\_WIN32) || defined(\_\_WIN32\_\_) || defined(WIN32)

#define \_CRT\_SECURE\_NO\_WARNINGS

#pragma warning(disable:4996)

#pragma warning(disable:6031)

#endif

#include <stdio.h>

#include <ctype.h>

int main() {

int c; //current simbol from the stream

int first = 1; //flag of the first letter in the word

int letter1 = 0; //first letter of the word

int counter = 0; //word counter

int endFlag = 1; //a flag that stops the application

//a global cycle that prevents the program from endting without the user's desire

do {

//a cycle for reading characters from a stream associated with the keyboard

printf("Please type down your words:\n");

while ((c = toupper(getchar())) != EOF){

if (c == ' ' || c == '.' || c == '\n' || c == ',') {

//separator found

first = 1;

}

else {

//letter found

if (first == 1) {

first = 0;

letter1 = c;

}

else {

//comparing not the first letter to the first one

if (letter1 == c) {

counter++;

letter1 = 0;

}

else {

letter1 = 0;

}

}

}

}

printf("The amount of words: %d", counter);

first = 0;

letter1 = 0;

counter = 0;

//suggesting to user to run program again or exit.

printf("\nPlease choose from the following:\n\t1 - Count again.\n\t2 - Exit.\n");

scanf("%d", &endFlag);

switch (endFlag) {

case 1: endFlag = 1; break;

case 2: endFlag = 0; break;

default: printf("Incorrect answer. The program will be closed."); endFlag = 0; break;

}

}while (endFlag); //if flag is 0, then exiting out of cycle

return 0;

}

1. **Выводы**

Для чтения символа из потока следует использовать функцию getchar().

Функция getchar() возвращает не char, а int потому что это необходимо для сравнения с символической константой EOF.

Символическая константа EOF используется для обозначения конца файла.

Таблица ASCII представляет из себя таблицу специальных кодов для 256 символов, упрощающую работу с символами в программах.

Escape-последовательность – это комбинация символов сообщающая программе о завершении действия.