

1. **Цель работы**

Изучение формата и правил описания символьных строк, а также методов обработки строковых данных.

1. **Постановка задачи**

Вариант 1: В символьной строке удалить все слова, начинающиеся с гласной буквы.

1. **Формализация задачи**

Для нахождения числа слов, у которых первая буква – гласная, необходимо разделять символы на слова и проверять первый символ слова. В число символов-разделителей входят: ' ', '.', '\n', '!', '?', ';', ':', '\_', '(', ')', '[', ']', '{', '}', '<', '>', '«', '»', '”', '’', '#', и ','. Буквы должны быть из латинского алфавита. Для решения задачи создаётся цикл проверки символов на то, являются ли они разделителями. Символы между разделителями – слова. Первый символ слова изменяет флаг first в 0 и сохраняемся как первая буква, которая затем проверяется на то, не является ли она гласной. Любой встретившийся разделитель вновь ставит влаг first в 1. После нахождения гласной буквы, она и все символы после неё до следующего разделителя заменяются на '\0'.

1. **Набор тестовых примеров**

|  |  |
| --- | --- |
| Вводимые слова | Вывод программы |
| Hello, I am here! | Hello, here! |
| No vowels to see here, just consonants. | No vowels to see here, just consonants. |

1. **Схема алгоритма**

Схема алгоритма программы представлена на рисунке 1.

Начало

N=1000; endFlag=1; size=N; mark; i=0; c[N]={0}; first=1; letter1='\0'; glasn[]="AEYUIO"; IsVowel = 0;

Ввод c[i]

Нет

Да

i < size

B

A

B

i < size

Да

Нет

A

c[i] - разделитель

Да

Нет

first = 1;

Да

Нет

first == 1

first = 0;

letter1 = c[i];

glasn[i]!=0

Нет

Да

letter1 == glasn[i]

Нет

Да

IsVowel = 0;

IsVowel = 1;

break;

C

D

D

C

IsVowel == 1

Нет

Да

mark = i;

IsVowel = 0;

Нет

c[mark] - разделитель

Да

c[mark] = '\0';

mark++;

first = 1;

letter1 = 0;

size = N;

IsVowel = 0;

Вывод результата

1 - Count again.

2 - Exit.

1

2

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

#define N 1000

#include <stdio.h>

#include <ctype.h>

int main() {

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

int size = N;

char c[N] = {0}; //current simbol from the stream

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

char letter1 = '\0'; //first letter of the word

char glasn[] = "AEYUIO"; //vowels array

int IsVowel = 0; //vowel flag

//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");

for (int i = 0; i < size; i++) {

scanf("%c", &c[i]);

if (i >= 1) {

if (c[i] == '\n') {

size = i;

i = size + 1;

}

}

}

//deleting '\n' in the beginning of a string

if (c[0] == '\n') {

for (int i = 0; i < size; i++) {

c[i] = c[i+1];

}

size--;

}

for (int i = 0; i < size; i++) {

if (c[i] == ' ' || c[i] == '.' || c[i] =='\0' || c[i] == '!' || c[i] == '?' || c[i] == ';' || c[i] == ':' || c[i] == '\_' || c[i] == '(' || c[i] == ')' || c[i] == '[' || c[i] == ']' || c[i] == '{' || c[i] == '}' || c[i] == '<' || c[i] == '>' || c[i] == '«' || c[i] == '»' || c[i] == '”' || c[i] == '’' || c[i] == '#' || c[i] == ',') {

//separator found

first = 1;

}

else {

//letter found

if (first == 1) {

first = 0;

letter1 = c[i];

//checking if the first letter is a vowel

for (int i = 0; glasn[i] != 0; i++)

{

if (toupper(letter1) == glasn[i])

{

IsVowel = 1;

break;

}

else {

IsVowel = 0;

}

}

//a word starting with a vowel is being erased

if (IsVowel == 1) {

int mark = i;

IsVowel = 0;

do {

c[mark] = '\0';

mark++;

} while (c[mark] != ' ' && c[mark] != '.' && c[mark] != '\0' && c[mark] != '!' && c[mark] != '?' && c[mark] != ';' && c[mark] != ':' && c[mark] != '\_' && c[mark] != '(' && c[mark] != ')' && c[mark] != '[' && c[mark] != ']' && c[mark] != '{' && c[mark] != '}' && c[mark] != '<' && c[mark] != '>' && c[mark] != '«' && c[mark] != '»' && c[mark] != '”' && c[mark] != '’' && c[mark] != '#' && c[mark] != ',' && mark < size);

}

}

}

}

printf("Your new sentence is: \'");

for (int i = 0; i < size; i++) {

printf("%c", c[i]);

}

printf("\'");

//resetting the variables

first = 1;

letter1 = 0;

size = N;

IsVowel = 0;

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

printf("\n\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**

Символьная строка в языке Си хранится в формате массива символов с завершающим нулём.

Различие ввода символьной строки при помощи функции scanf() и gets() заключается в том, что scanf() заканчивает принимать ввод при встрече с пробелами, символом новой строки или EOF, а gets() рассматривает пробелы как часть входной строки и заканчивает ввод при встрече с новой строки или EOF.

Правильно описать массив для хранения символьной строки следует следующим образом: char c[N] = {0};

Символьная строка длины 99 может храниться в массиве, описанном как char line[100], поскольку последний символ будет \0.

Библиотечная функция, определяющая длину символьной строки, должна работать по принципу прохождения по массиву и увеличения счётчика символов, пока не встретится EOF.