Отчёт по лабораторной работе №6

Тема: Операции с массивами. Работа с псевдослучайными последовательностями(ПСП).

Задание (вариант 8): С помощью ПСП сформировать двухмерный MxN массив из текстовых элементов. В качестве элементов использовать слова из K…L символов A…Z. Найти все слова, в которых все нечетные буквы – гласные. Вывести на экран сформированный массив, выделив цветом или другим способом указанные в варианте элементы, и отдельно – результат подсчета этих элементов.

Выполнил:

Студент

Вискунов Дмитрий Юрьевич

КОБ-22-1с

**Постановка задач.**

1.Создание массива по заданным параметрам заполненного текстовыми элементов случайной последовательности.

2.Разработка метода проверки условия (все нечетные буквы в слове – гласные.), вывод массива и результата подсчета подходящих под условие элементов.

**Структура программы.**

1.Подключение библиотек. <iostream> - библиотека ввода вывода данных. <string> - библиотека для работы со строками используется для создания, изменения и удаления строки. <ctime> - библиотека для работы со временем в данной программе используется для создания новой последовательности в функции srand().

2.Объявление переменных. Создание строки alphabet в которой соответственно содержатся все символы латинского алфавита. Создание одномерного массива vowels\_alphabet в котором содержатся все гласные буквы латинского алфавита.

3. Использование функции srand(time(NULL)) данная функция будет генерировать новую последовательность функции rand при каждом изменении системного времени.

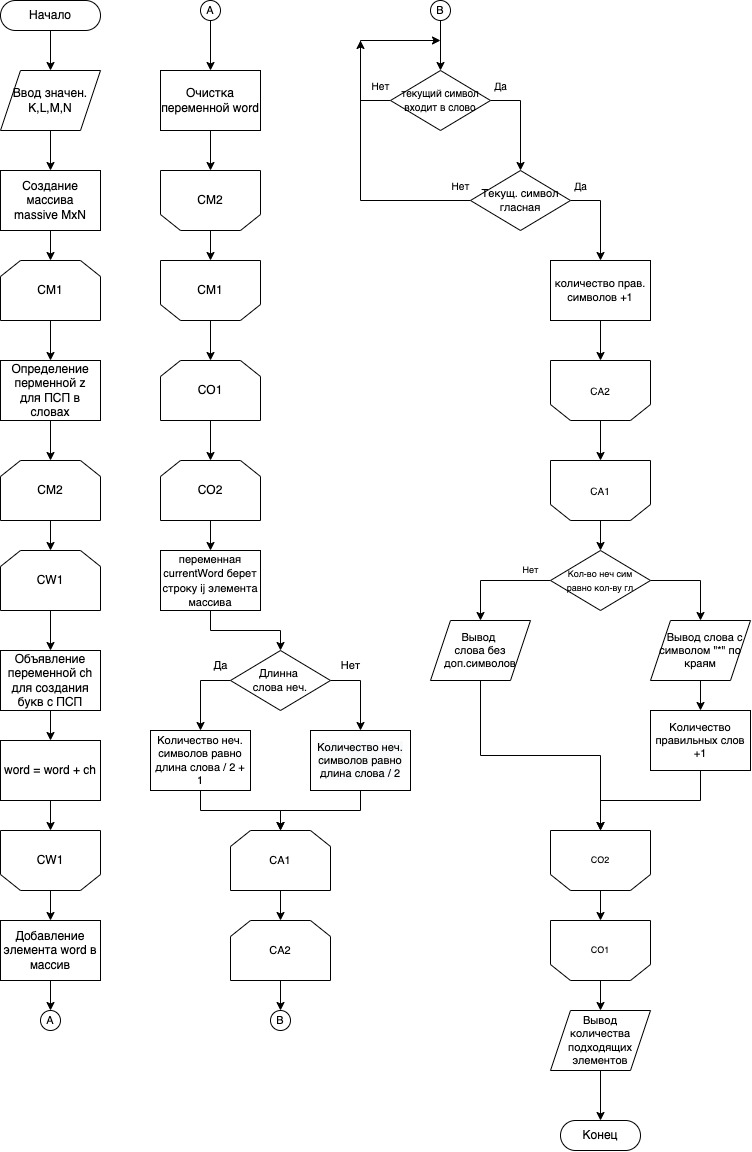
4. Ввод значений для переменных k,l,m,n.

5. Создание двухмерного массива massive[m][n] размеры которого соответственно MxN. Заполнение этого массива выполняется через цикл.

6. Генерация случайных символов и соответственно случайных слов со случайным количеством букв в диапазоне K..L с помощью функции rand().

7. Отображение двухмерного массива с проверкой условия задания, вывод массива осуществляется при помощи циклов. На данном этапе вычисляется количество нечетных букв, данное число понадобится в дальнейшем для проверки условия. Проверка условия и проверка не вышел ли алгоритм за пределы слова. Вывод элементов массива, если данный элемент соответствует условию, то он выводиться с символом «\*» в начале и конце, а так же счетчик соответствующих условию элементов увеличивается на 1. Для правильного отображения массива используется функция cout.width(l) которая выделяет на каждый элемент массива размер «l» где «l»– самый большой элемент массива. Вывод количества соответствующих элементов.

**Графическое описание алгоритма**



**Код программы:**

#include <iostream>

#include <ctime>

using namespace std;

int k,l,m,n;

int i,j;

int right\_elements;

string word;

int main() {

srand(time(NULL));

string alphabet = "abcdefghijklmnopqrstuvwxyz";

char vowels\_alphabet[6] = {'a','e','i','o','u','y'};

cout << "Введите значение K:" << endl;

cin >> k;

cout << "Введите значение L:" << endl;

cin >> l;

cout << "Введите значение M:" << endl;

cin >> m;

cout << "Введите значение N:" << endl;

cin >> n;

string massive[m][n];

for (i=0;i<m;i++){ // создаем массив из рандомных слов состоящих из символов alphabet

int z = k + rand() % (l-k); // рандомизация количества букв

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

for(int iter = 0;iter<z;iter++){

char ch = alphabet[rand() % alphabet.size()];

word = word + ch;

}

massive[j][i] = word;

word = "";

}

}

for (i = 0;i<m;i++){ // выводим массив с проверкой условия.

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

string currentWord = massive[i][j];

int len = currentWord.size();

int oddsymbols;

int right\_symbols = 0;

if(len % 2 == 1){ // вычисляем количество нечетных символов

oddsymbols = len/2 + 1;

}else{

oddsymbols = len/2;

}

for(int index = 0; index <= len; index = index + 2){

for(int ArraySymbol = 0; ArraySymbol < 6;ArraySymbol++){

if(index <= len){

if(vowels\_alphabet[ArraySymbol] == currentWord[index]){right\_symbols++;}

}

}

}

if(right\_symbols == oddsymbols){

cout.width(l);

cout << "\*" << massive[i][j] <<"\*";

right\_elements++;

}else{

cout.width(l);

cout <<"\t" << massive[i][j];

}

}

cout <<"\n";}

cout <<endl << "Всего подходящих под условие элементов:"<<right\_elements<<endl;

}

**Результат работы программы**

