**Лабораторная работа №5**

**«ОБРАБОТКА ОДНОМЕРНЫХ ДИНАМИЧЕСКИХ МАССИВОВ С ПОМОЩЬЮ ФУНКЦИЙ»**

**Цель работы**: Изучение особенностей представления и обработки одномерных массивов в языках С/С++ с учетом связи указателей и массивов. Получение практических навыков реализации алгоритмов обработки одномерных динамических массивов средствами языков С/С++. Исследование особенностей обработки одномерных динамических массивов.**1. Вариант задания**

Последовательность состоит из символов. Необходимо разделить символы последовательности так, чтобы в левой ее части располагались гласные буквы, а в правой – согласные.

Упорядочить элементы последовательности ap, ap+1, …, aq по возрастанию, используя алгоритм сортировки методом прямого выбора.

**2. Алгоритм программы**

Программа состоит из нескольких функций которые вызываются в основной функции main. Для реализации данной программы были сделаны несколько отдельных функций. Схемы всех функций будут приведены ниже.



Рисунок 1 – структурная схема функции ввода (слева) и функции вывода (справа) последовательности



Рисунок 2 – Структурная схема функции переноса гласных в левую часть.



Рисунок 3 – Структурная схема алгоритма сортировке методом прямого выбора.

**3. Код программы**

// LR5.cpp : Этот файл содержит функцию "main". Здесь начинается и заканчивается выполнение программы.

//

#pragma warning(disable : 4996)

#include <Windows.h>

#include<stdio.h>

#include <stdlib.h>

#include <string.h>

const char glas[] = "aeiouy";//массив гласных

char\* in\_pos(char\* pos, int n);// Ввод последовательности

void out\_pos(char\* pos, int n);// Вывод последовательности

char\* del\_pos(char\* pos, int n);//деление гласных и согласных

char\* sort\_pos(char\* pos);//Сортировка последовательности

main()

{

SetConsoleCP(1251); // Задаем таблицу символов для консоли.

SetConsoleOutputCP(1251);

printf("Введите размер последовательности -->");

int n; scanf("%d", &n); // считываем колличество элементов в последовательности

char\* pos = in\_pos(&pos, n);

printf("\nВведенная последовательность :");

out\_pos(pos, n); //вызов вывода последовательности

pos = del\_pos(pos, n); //вызов функции деления последовательности по заданному правилу

printf("\nПреобразованная последовательность :");

out\_pos(pos,n); // вызов вывода последовательности

pos = sort\_pos(pos); //вызов сортировки

printf("\nОтсортированная пооследовательность :");

out\_pos(pos, n);// вызов вывода последовательности

free (pos);

return 0;

}

char\* in\_pos(char\* pos, int n) //функция ввода последовательности

{

pos = (char\*)malloc(sizeof(char) \* (n+1)); //чистим память под последовательность

printf("Введите последовательность --> ");

scanf("%s", pos); //считываем последовательность в виде строки

return pos;

}

void out\_pos(char\* pos,int n)// Вывод последовательности

{

for (int i = 0; i < n; i++) { //цикл вывода

printf("%c", pos[i]); //вывод элемента

}

}

char\* del\_pos(char\* pos, int n)

{

int counter=0; //счётчик текущего положения

char\* temp = (char\*)malloc(sizeof(char) \* (n )); //выделяем память под копию строки

memmove(temp, pos, n); //делаем копию строки для предотвращения утери данных

for (int i = 0; i < n; i++) //цикл поиска гласных

{

for (int j = 0; j < 6; j++) //проверка по гласным

{

if (temp[i] == glas[j]) //если нашли гласную

{

pos[counter] = temp[i]; //записываем гласную

counter++;//увеличиваем позицю

break;//завершаем проверку

}

}

}

for (int i = 0; i < n; i++) //второй цикл поиска не гласных

{

int tip = 0;//если тип = 0 , то это согласная буква , иначе гласная

for (int j = 0; j < 6; j++) //проверка , гласный ли текущий символ

{

if (temp[i] == glas[j])

{

tip = 1;

break;

}

}

if (!tip) { // если он не гласный , то записываем его

pos[counter] = temp[i];

counter++; // переводим позицию

}

}

free(temp); //очищаем занимаемую массивом temp память

return pos; // возвращаем значения

}

char\* sort\_pos(char\* pos)

{

printf("\nВведите начальный индекс сортировки -->");

int p; scanf("%d", &p);

printf("\nВведите конечный индекс сортировки -->");

int q; scanf("%d", &q);

char temp;

for (int j = q; j > 1; j--) //сортировка левой части

for (int i = p; i < j ; i++)// сравнение двух соседних элементов

if (pos[i] > pos[i + 1])

{

temp = pos[i]; // перестановка элементов

pos[i] = pos[i + 1];

pos[i + 1] = temp;

}

return pos;

}

**4.Результаты тестирования и отладки программы**

Для отладки и тестирования программы будут использованные заранее просчитанные в ручную тестовые примеры. Вариант их вида будет приведен в таблице ниже. Необходимо учитывать что программа учитывает только английский язык, так же размер последовательности и сама последовательность должна совпадать (количество элементов последовательности = ее размеру)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Последовательность | Количество символов | Последовательность с переносом гласных | P | Q | Отсортированная последовательность |
| nhtriseadjklpoybvcxz | 20 | ieaoynhtrsdjklpbvcxz | 4 | 16 | ieaobdhjklnprstvycxz |
| hmgndaiopasdcxrkeqfbvrilyqwjmn | 30 | aioaeiyhmgndpsdcxrkqfbvrlqwjmn | 10 | 30 | aioaeiyhmgbcddfjklmnnpqqrrsvw |
| aioaeiyhmgfbvrilyqwjklpoybvcxzieaoynhtrs | 40 | aioaeiyiyoyieaoyhmgfbvrlqwjklpbvcxznhtrs | 20 | 35 | aioaeiyiyoyieaoyhmgfbbcjkllnpqrvvwxzhtrs |

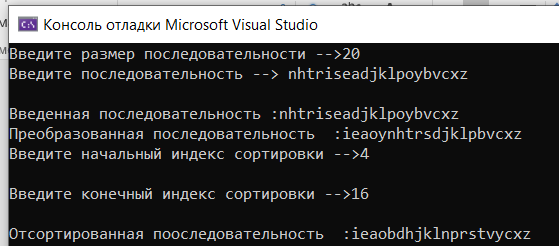
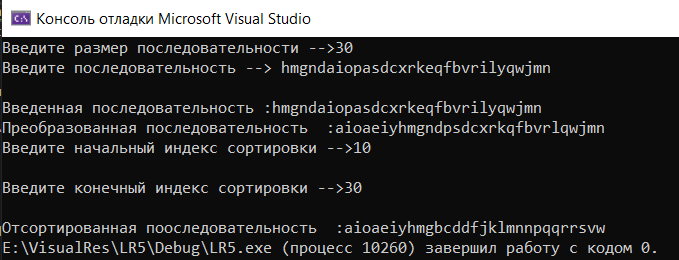
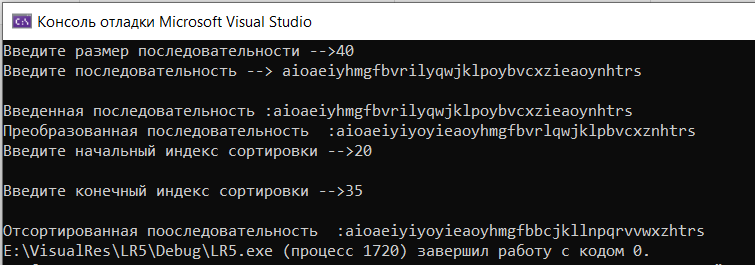
  

Рисунок 4 – Выполнение тестовых примеров в программе

В результате тестирования программы была выполнена проверка правильности выполнения программы. Результаты полученные при выполнении программы и тестовые примеры полностью совпадают. Программа работает верно.

**Вывод**

В ходе выполнения данной лабораторной работы были изучены особенности представления и обработки одномерных массивов в языках С/С++ с учетом связи указателей и массивов. Получены практические навыки реализации алгоритмов обработки одномерных динамических массивов средствами языков С/С++. Были про исследованы особенности обработки одномерных динамических массивов. Так же были изучены функции работы с памятью.