Министерство образования и науки Российской Федерации

Федеральное государственное бюджетное образовательное учреждение

высшего образования

**«Пермский национальный исследовательский**

**политехнический университет»**

Кафедра «Информационные технологии и автоматизированные системы»

**О Т Ч Ё Т**

**по лабораторной работе №10**

Дисциплина: основы алгоритмизации и программирования

Тема: “Динамические массивы”

Вариант 16

Выполнил:

студент группы ИВТ-20-2Б Тедеев А.З.

Проверила: доцент кафедры ИТАС

Полякова О.А.

Пермь, 2021

**Цель задачи**

Организация динамических массивов.

**Постановка задачи**

Сформировать массив строк. Удалить из него самую короткую строку.

**Анализ задачи**

1. int rows=-1; string\* arr = new string[rows];

Rows – количество строк, arr – массив строк.

2. Используются в качестве параметров для функций.

3. Задача решается путём организации двух динамических массивов: в главной функции организуется динамический массив строк, а в функции ChangeArr() типа void организуется дополнительный динамический массив tmp с размерностью rows-1

void ChangeArr(string\*& arr, int& rows)

{

string\* tmp = new string[rows-1];

...

}

4. Ввод строк в массив arr организуется в функции ShowArr() через функцию getline(). Ввод количества строк организуется в функции main() через команду cout.

5. Чтобы пользователь не ввёл отрицательное количество строк, использовать цикл while

while (rows < 0) {

cout << "Введите количество строк: ";

cin >> rows;

}

Для заполнения массива используется функция FillArr().

void FillArr(string\* arr, int rows)

{

while (cin.get() != '\n');

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

cout << "Введите строку: ";

getline(cin, arr[i]);

}

cout << endl;

}

Для вывода массива строк в консоль использовать функцию ShowArr().

void ShowArr(string\* arr, int rows)

{

cout << "Вывод массива: "<< endl;

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

cout << arr[i] << endl;

}

cout << "---------------------";

}

Удаление самой короткой строки происходит в функции ChangeArr() . В цикле for происходит поиск самой короткой строки через функцию length(), в переменную ind запоминается индекс самой короткой строки в массиве.

string min = arr[0];

int min1=arr[0].length(), ind=0;

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

if (min.length()>arr[i].length()) {

min = arr[i];

min1 = arr[i].length();

ind = i;

}

}

После данного цикла заполняем дополнительный массив tmp в двух циклах. В первом из них заполнение ведётся до номера самой короткой строки. Во втором от номера самой короткой строки до rows-1. Затем необходимо освободить память для указателя arr и присвоить указатель на массив tmp.

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

tmp[i] = arr[i];

}

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

tmp[i] = arr[i + 1];

}

delete[] arr;

rows--;

arr = tmp;

Через функцию ShowArr() выводим изменнёный массив и очищаем память через delete[] arr.

**Код**

#include <iostream>

#include <string>

using namespace std;

void FillArr(string\* arr, int rows);

void ShowArr(string\* arr, int rows);

void ChangeArr(string\*& arr, int& rows);

int main()

{

system("chcp 1251>nul");

setlocale(LC\_ALL, "ru");

int rows=-1;

while (rows < 0) {

cout << "Введите количество строк: ";

cin >> rows;

}

string\* arr = new string[rows];

FillArr(arr, rows);

ShowArr(arr, rows);

ChangeArr(arr, rows);

cout << "\n\n";

ShowArr(arr, rows);

delete[] arr;

}

void FillArr(string\* arr, int rows)

{

while (cin.get() != '\n');

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

cout << "Введите строку: ";

getline(cin, arr[i]);

}

cout << endl;

}

void ShowArr(string\* arr, int rows)

{

cout << "Вывод массива: "<< endl;

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

cout << arr[i] << endl;

}

cout << "---------------------";

}

void ChangeArr(string\*& arr, int& rows)

{

string\* tmp = new string[rows-1];

string min = arr[0];

int min1=arr[0].length(), ind=0;

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

if (arr[i].length()<min.length()) {

min = arr[i];

min1 = arr[i].length();

ind = i;

}

}

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

tmp[i] = arr[i];

}

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

tmp[i] = arr[i + 1];

}

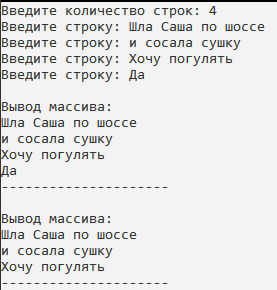
delete[] arr;

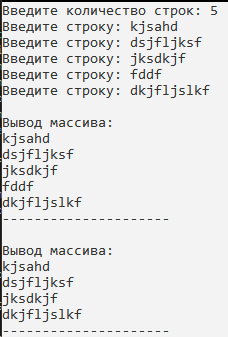
rows--;

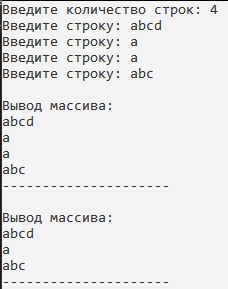
arr = tmp;

}

**Скриншоты**







**Блок-схема**

