НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ імені ІГОРЯ СІКОРСЬКОГО»

Інститут прикладного системного аналізу

Кафедра математичних методів системного аналізу

Звіт

про виконання лабораторної роботи № 9 з дисципліни «Алгоритмізація та програмування»

Виконав:

студент I курсу, групи КА-07

Лещинський Богдан Денисович

Прийняв:

Київ — 2020

Завдання.

Ввести масив записів, які містять відомості-назви і адреси інститутів, програма видаляє записи, заповнені тільки пробілами (пропусками).

Аналіз умови задачі.

Вхідна змвнна одна – number, що задає кількість структур Institution у початковому динамічному масиві структур institutions1. Тип даних number – цілі числа (int).

Вихідних змінних дві - institutions1 та institutions2. Тип данних змінних – динамічні массиви структур даних Insitution.

Виходячи з умови задачі необхідно розбити алгоритм на такі базові підзадачі: створення, відповідно до умови завдання, структури данних з полями name, address, name, address надання користувачеві можливості вводу бажаної кількості структур у початковому динамічному масиві структур і подальше виділення пам’яті під цей масив.

Після цього, заповнити перший масив institutions1 випадковими структурами даних, беручи дані для них з двох масивів можливих назв та адрес – possibleNames та possibleAdresses. Заповнивши масив, з нього, згідно умови поставленої задачі, потрібно видалити ті записи, що містять у собі лише пробіли (пропуски).

Порахувавши кількість структур Insitution, що не задовільняють критерій видалення, за допомогою цикла for, створимо новий динамічний масив структур institutions2, під який виділимо рівно стільки пам’яті, скільки структур у початковому масивів institutions1 не підпадають під критерій видалення, наданий умовою (їх кількість була порахована циклом for та збережена у змінній counter, що має цілоциельний тип даних іnt). У новий масив перенесемо відповідно лише структури, що не підпадають під видалення .

Код програми :

#include <iostream>

#include <cstdlib>

#include <string>

#include <ctime>

#include <locale.h>

using namespace std;

struct Institution {

string name;

string adress;

};

int main() {

srand(time(NULL));

setlocale(LC\_ALL, "RUS");

string possibleNames[] = { "КНУ iм.Тараса Шевченка", "НТУУ КПI iм.Ігоря Сiкорського",

"Могилянка", "Харчових технологiй", "ДУТ", "Львiвська Полiтехнiка", "УКУ", " " };

string possibleAdresses[] = { "вул.Шолуденка 19", "пр.Перемоги 15", "вул.Льва Толстого 17", "Європейська площа 1",

"пр.Степана Бандери 16", "вул. Лейтенандта Шмiдта", "вул. Волл 0", " " };

cout << "--Enter number of educational institutions: \n";

int number;

cin >> number;

Institution \*institutions1 = new Institution[number]; // creating a dynamic array of structures

cout << "----Initially generated array of educational insitutions: \n";

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

institutions1[i] = { possibleNames[rand() % 8],possibleAdresses[rand() % 8] };

cout << institutions1[i].name << " " << institutions1[i].adress<<endl;

}

cout << endl;

cout << "----Edited array of educational insitutions: \n";

int counter = 0;

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

if (institutions1[i].name != " " && institutions1[i].adress != " ") {

counter++;

}

}

Institution \*institutions2 = new Institution[counter];

int p = 0;

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

if(institutions1[i].name != " " && institutions1[i].adress != " "){

institutions2[p] = institutions1[i];

cout << institutions2[p].name << " " << institutions2[p].adress << endl;

p++;

}

}

delete[] institutions1;

delete[] institutions2;

return 0;

}

Результат виконання програми:

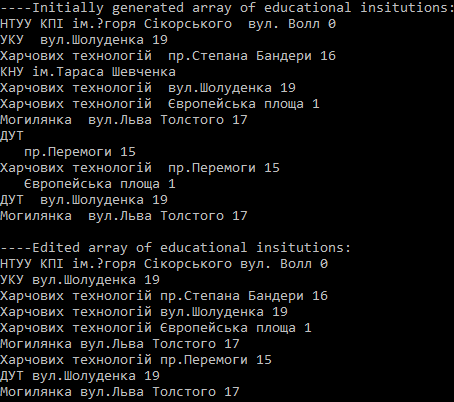


Рисунок 2 - результат виконання програми

Висновки.

У результаті виконання даної лабораторної роботи було розроблено програму, яка видаляє з масиву структур структури, за певним критерієм. Було створено структуру даних, що містить у собі поля name, address, а також динамічний масив описаних структур за допомогою оператора виділення пам’яті new (під кінець програми виділена пам’ять видаляється за допомогою оператора delete). Після заповнення масиву випадковими структурами даних, було визначено кількість структур, що не підлягають видаленню за умовою завдання, та створено новий динамічний масив для їх зберігання. За допомогою цикла for у новий масив структур були перенесені структури зі старого, які, як вже було зазначено, не потрібно було видаляти за умовою.