ЗВІТ

про виконання лабораторної роботи № 9.2.B

«Впорядкування та бінарний пошук в масиві структур»

з дисципліни

«Алгоритмізація та програмування»

студентки групи РІ-11

Пятницької Вікторії Володимирівни

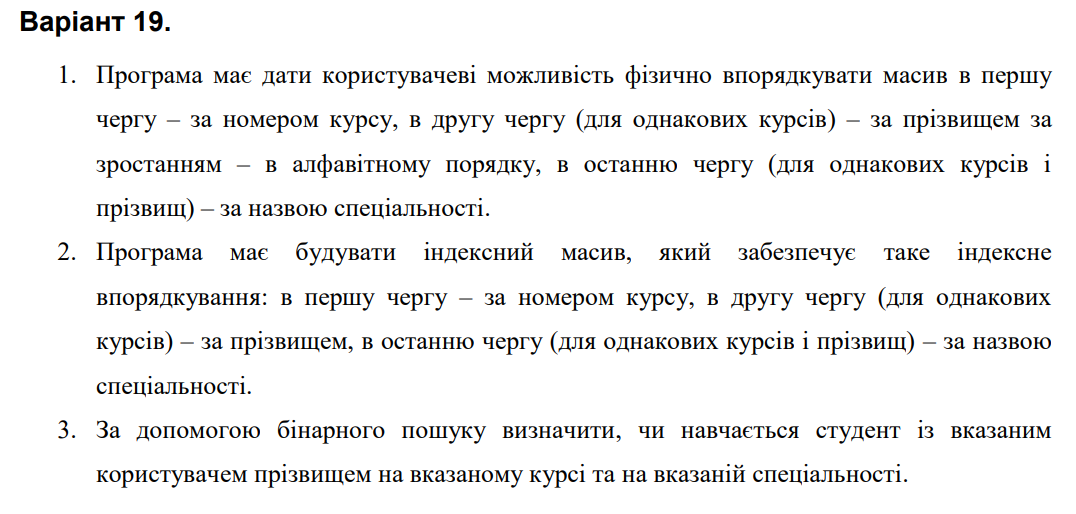
**Мета роботи**

Навчитися впорядковувати масив структур з об’єднаннями. Навчитися здійснювати

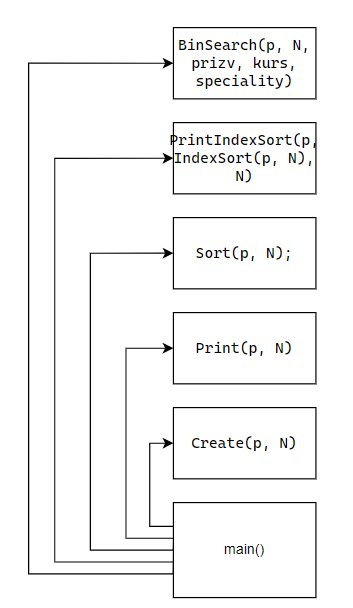
фізичне та індексне впорядкування. Навчитися здійснювати бінарний пошук у фізично чи

індексно впорядкованому масиві.

**Умова завдання:**



**Структурна схема програми:**

****

**Текст програми:**

#include <iostream>

#include <iomanip>

#include <string>

#include <Windows.h>

using namespace std;

enum Spec { KN, IT, MATH, PHYSIC, WORK };

string specStr[] = { "Комп'ютерні науки", "Інформатика", "Математика та економіка", "Фізика та інформатика", "Трудове навчання" };

struct Student

{

string prizv;

Spec spec;

int kurs;

struct

{

int physic;

int math;

};

union

{

int IT;

int nummet;

int pedagogy;

};

};

void Create(Student\* p, const int N)

{

int spec;

int kurs;

for (int i = 0; i < N; i++)

{

cout << "Студент № " << i + 1 << ":" << endl;

cin.get();

cin.sync();

cout << " Прізвище: "; getline(cin, p[i].prizv);

cout << " Курс: ";

cin >> p[i].kurs;

cout << " Спеціальність (0 - Комп’ютерні науки, 1 - Інформатика, 2 - Математика та економіка, 3 - Фізика та інформатика, 4 - Трудове навчання): ";

cin >> spec;

cout << " Оцінка з фізики : "; cin >> p[i].physic;

cout << " Оцінка з математики : "; cin >> p[i].math;

p[i].spec = (Spec)spec;

switch (p[i].spec)

{

case KN:

cout << " Оцінка з програмування : "; cin >> p[i].IT;

break;

case IT:

cout << " Оцінка з чисельних методів : "; cin >> p[i].nummet;

break;

case MATH:

case PHYSIC:

case WORK:

cout << " Оцінка з педагогіки : "; cin >> p[i].pedagogy;

break;

}

cout << endl;

}

}

void Print(Student\* p, const int N)

{

cout << "======================================================================================================================================================================="

<< endl;

cout << "| № | Прізвище | Курс | Cпеціальність | Оцінка з фізики | Оцінка з математики | Оцінка з програмування | Оцінка з чисельних методів | Оцінка з педагогіки |"

<< endl;

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

<< endl;

cout << fixed;

for (int i = 0; i < N; i++)

{

cout << "| " << setw(2) << i + 1;

cout << "| " << setw(9) << p[i].prizv;

cout << "| " << setw(5) << p[i].kurs;

cout << "| " << setw(25) << specStr[p[i].spec];

cout << "| " << setw(16) << p[i].physic;

cout << "| " << setw(20) << p[i].math;

switch (p[i].spec)

{

case KN:

cout << "| " << setw(23) << p[i].IT << "|" << setw(29) << "|" << setw(22) << "|";

break;

case IT:

cout << "| " << setw(24) << "|" << setw(28) << p[i].nummet << "|" << setw(22) << "|";

break;

case MATH:

case PHYSIC:

case WORK:

cout << "| " << setw(24) << "|" << setw(29) << "|" << setw(21) << p[i].pedagogy << "|" << endl;

break;

}

cout << endl;

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

cout << endl;

}

}

void Sort(Student\* p, const int N)

{

Student tmp;

for (int i = 0; i < N - 1; i++)

for (int j = 0; j < N - i - 1; j++)

{

if (

(p[j].kurs > p[j + 1].kurs)

||

(p[j].kurs == p[j + 1].kurs &&

p[j].prizv > p[j + 1].prizv)

||

(p[j].kurs) == p[j + 1].kurs &&

p[j].prizv == p[j + 1].prizv &&

p[j].spec > p[j + 1].spec);

{

tmp = p[j];

p[j] = p[j + 1];

p[j + 1] = tmp;

}

}

}

int\* IndexSort(Student\* p, const int N)

{

int\* I = new int[N];

for (int i = 0; i < N; i++)

I[i] = i;

int i, j, value;

for (i = 1; i < N; i++)

{

value = I[i];

for (j = i - 1;

j >= 0 && ((p[j].kurs > p[j + 1].kurs)

||

(p[j].kurs == p[j + 1].kurs &&

p[j].prizv > p[j + 1].prizv)

||

(p[j].kurs) == p[j + 1].kurs &&

p[j].prizv == p[j + 1].prizv &&

p[j].spec > p[j + 1].spec);

j--)

{

I[j + 1] = I[j];

}

I[j + 1] = value;

}

return I;

}

void PrintIndexSort(Student\* p, int\* I, const int N)

{

cout << "======================================================================================================================================================================="

<< endl;

cout << "| № | Прізвище | Курс | Cпеціальність | Оцінка з фізики | Оцінка з математики | Оцінка з програмування | Оцінка з чисельних методів | Оцінка з педагогіки |"

<< endl;

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

<< endl;

cout << fixed;

for (int i = 0; i < N; i++)

{

cout << "| " << setw(2) << i + 1;

cout << "| " << setw(9) << p[i].prizv;

cout << "| " << setw(5) << p[i].kurs;

cout << "| " << setw(25) << specStr[p[i].spec];

cout << "| " << setw(16) << p[i].physic;

cout << "| " << setw(20) << p[i].math;

switch (p[i].spec)

{

case KN:

cout << "| " << setw(23) << p[i].IT << "|" << setw(29) << "|" << setw(22) << "|";

break;

case IT:

cout << "| " << setw(24) << "|" << setw(28) << p[i].nummet << "|" << setw(22) << "|";

break;

case MATH:

case PHYSIC:

case WORK:

cout << "| " << setw(24) << "|" << setw(29) << "|" << setw(21) << p[i].pedagogy << "|" << endl;

break;

}

cout << endl;

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

cout << endl;

}

}

int BinSearch(Student\* p, const int N, const string prizv, const int kurs, const int speciality)

{

int L = 0, R = N - 1, m;

do {

m = (L + R) / 2;

if (p[m].prizv == prizv && p[m].kurs == kurs && p[m].spec == speciality)

return m;

if ((p[m].kurs < kurs)

||

(p[m].kurs == kurs &&

p[m].prizv < prizv)

||

(p[m].kurs) == kurs &&

p[m].prizv == prizv &&

p[m].spec < speciality)

{

L = m + 1;

}

else

{

R = m - 1;

}

} while (L <= R);

return -1;

}

int main()

{

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

int N;

cout << "Введіть число студентів: "; cin >> N;

Student\* p = new Student[N];

string prizv;

int kurs;

Spec spec;

int speciality;

int found;

int menuItem = 0;

do

{

cout << "Виберіть дію:" << endl

<< "1. Заповнити дані" << endl

<< "2. Вивести на екран дані" << endl

<< "3. Фізичне впорядкування даних" << endl

<< "4. Індексне впорядкування даних" << endl

<< "5. Бінарний пошук студента у масиві за вказаним прізвищем, курсом та спеціальністю" << endl

<< "0. Завершити роботу програми" << endl;

cin >> menuItem; cout << endl;

switch (menuItem)

{

case 1:

Create(p, N);

cout << endl;

break;

case 2:

Print(p, N);

cout << endl;

break;

case 3:

cout << "Фізичне впорядкування даних:" << endl;

Sort(p, N);

Print(p, N);

cout << endl;

break;

case 4:

PrintIndexSort(p, IndexSort(p, N), N);

break;

case 5:

cout << "Введіть прізвище студента:"; cin >> prizv;

cin.sync();

cout << "Введіть курс, на якому навчається студент:";

cin >> kurs;

cin.get();

cin.sync();

cout << "Введіть спеціальність, на якій навчається студент:"; cin >> speciality;

cout << endl;

speciality = Spec(speciality);

if ((found = BinSearch(p, N, prizv, kurs, speciality)) != -1)

cout << "Знайдено студента за вказаними критеріями в позиції: " << found + 1 << endl;

else

cout << "Такого студента не знайдено."; cout << endl;

break;

case 0:

break;

default:

cout << "Ви ввели помилкове значення!"

"Слід ввести число - номер вибраного пункту меню" << endl;

}

} while (menuItem != 0);

cin.get();

return 0;

}

**Посилання на git-репозиторій з проектом:**

[**https://github.com/viksi01/lab.9.2**](https://github.com/viksi01/lab.9.2)

**Результати unit-тесту:**

#include "pch.h"

#include "CppUnitTest.h"

#include "../lab\_9.2.B/lab\_9.2.B.cpp"

using namespace Microsoft::VisualStudio::CppUnitTestFramework;

namespace UnitTest92B

{

TEST\_CLASS(UnitTest92B)

{

public:

TEST\_METHOD(TestMethod1)

{

Student\* p = new Student;

p->prizv = "B";

p->kurs = 3;

p->spec = KN;

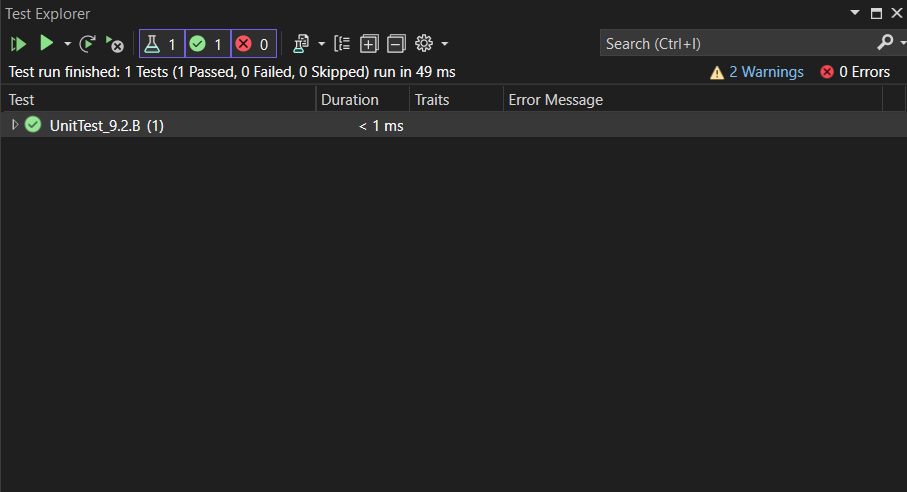
int t = BinSearch(p, 1, "B", 3, KN);

Assert::AreEqual(t, 0);

}

};

}

****

**Висновки:**

Отже, виконавши цю лабораторну роботу я навчилася впорядковувати масив структур з об’єднаннями, здійснювати фізичне та індексне впорядкування, здійснювати бінарний пошук у фізично чи індексно впорядкованому масиві.