ЗВІТ

про виконання лабораторної роботи № <9.2>

*«Впорядкування та бінарний пошук в масиві структур»*

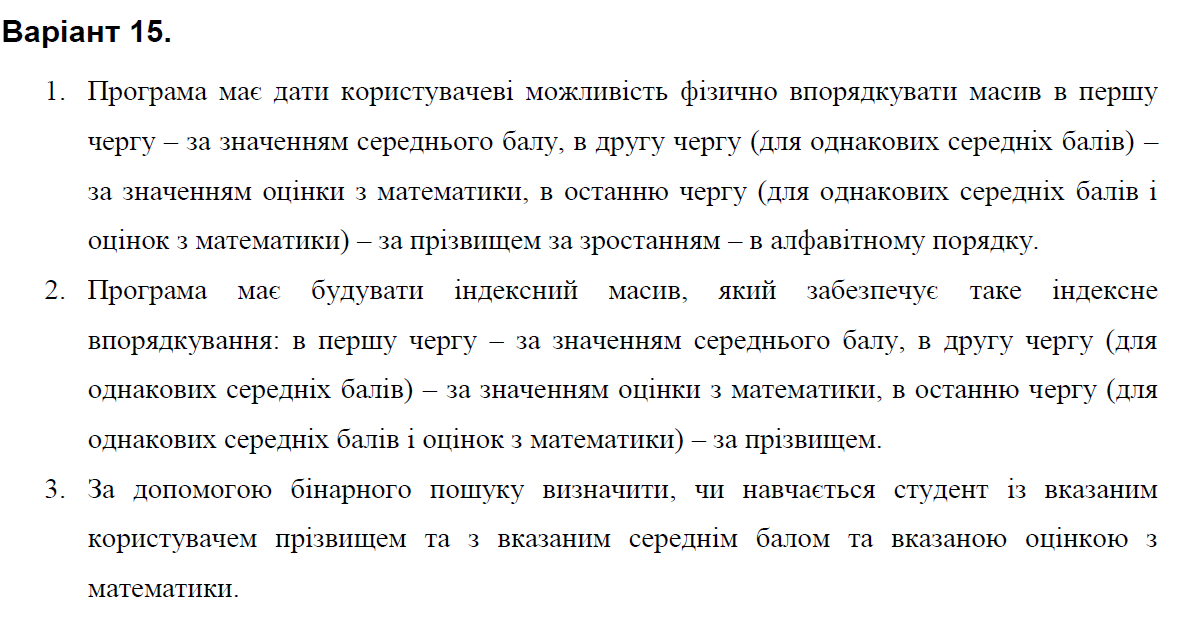
з дисципліни

«Алгоритмізація та програмування»

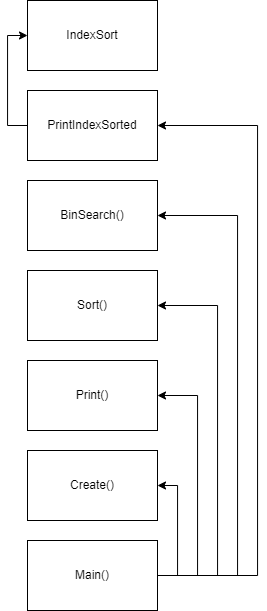
студента групи ІК-11

< *Снігура Стефана Андрійовича* >

**Умова завдання:**

****

**Структурна схема програми :**

****

**Текст програми:**

#include <iostream>

#include <fstream>

#include <iomanip>

#include <string>

#include <Windows.h> // забезпечення відображення кирилиці

using namespace std;

struct Student

{

string prizv;

unsigned kurs;

string specialnist;

unsigned fizyka;

unsigned matematyka;

unsigned informatyka;

};

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

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

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

int BinSearch(Student\* p, const int N, const string prizv, double MathG, double average);

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

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

int main()

{

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

int N;

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

Student\* p = new Student[N];

double proc, avg, MathG, average;

string prizv;

int found;

char filename[100];

int menuItem;

do {

cout << endl << endl << endl;

cout << "Виберіть дію:" << endl << endl;

cout << " [1] - введення даних з клавіатури" << endl;

cout << " [2] - вивід даних на екран" << endl;

cout << " [3] - фізичне впорядкування даних" << endl;

cout << " [4] - бінарний пошук студента за посадою та прізвищем" << endl;

cout << " [5] - індексне впорядкування та вивід даних" << endl;

cout << " [0] - вихід та завершення роботи програми" << endl << endl;

cout << "Введіть значення: "; cin >> menuItem;

cout << endl << endl << endl;

switch (menuItem)

{

case 1:

Create(p, N);

break;

case 2:

Print(p, N);

break;

case 3:

Sort(p, N);

break;

case 4:

cout << "Введіть ключі пошуку:" << endl;

cin.get(); // очищуємо буфер клавіатури – бо залишаються символи

cin.sync(); // "кінець рядка", які не дають ввести наступний літерний рядок

cout << "Введіть прізвище: "; getline(cin, prizv);

cout << endl;

cout << "Введіть оцінку з математики: ";

cin >> MathG;

cout << "Введіть середній бал: ";

cin >> average;

if ((found = BinSearch(p, N, prizv, MathG, average)) != -1)

cout << "Знайдено студента в позиції " << found + 1 << endl;

else

cout << "Шуканого студента не знайдено" << endl;

break;

case 5:

PrintIndexSorted(p, IndexSort(p, N), N);

break;

default:

cout << "Ви ввели помилкове значення! "

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

}

} while (menuItem != 0);

return 0;

}

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

{

int Predmet;

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

{

int n = 0;

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

cin.get();

cin.sync();

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

cout << " курс: "; cin >> p[i].kurs;

cout << " спеціальність "; cin >> p[i].specialnist;

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

cout << endl;

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

cout << endl;

cout << " Оцінка з інформатики: "; cin >> p[i].informatyka;

cout << endl;

}

}

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

{

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

<< endl;

cout << "\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_| Бали |"

<< endl;

cout << "| № | Прізвище | Курс | Спеціальність | Фізика | Математика | Інформатика |"

<< endl;

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

<< endl;

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

{

cout << "| " << setw(3) << right << i + 1 << " ";

cout << "| " << setw(11) << left << p[i].prizv

<< "| " << setw(4) << right << p[i].kurs << " "

<< "| " << setw(14) << left << p[i].specialnist

<< "| " << setw(8) << right << p[i].fizyka << " "

<< "| " << setw(12) << right << p[i].matematyka << " "

<< "| " << setw(11) << right << p[i].informatyka << " |" << endl;

}

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

<< endl;

cout << endl;

}

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

{

Student tmp;

double avg1, avg2;

for (int i0 = 0; i0 < N - 1; i0++)

for (int i1 = 0; i1 < N - i0 - 1; i1++) {

avg1 = (p[i1].fizyka + p[i1].matematyka + p[i1].informatyka) / 3.0;

avg2 = (p[i1 + 1].fizyka + p[i1 + 1].matematyka + p[i1 + 1].informatyka) / 3.0;

if (avg1 < avg2

||

(avg1 == avg2 &&

p[i1].matematyka < p[i1 + 1].matematyka)

||

(avg1 == avg2 &&

p[i1].matematyka == p[i1 + 1].matematyka &&

p[i1].prizv < p[i1 + 1].prizv))

{

tmp = p[i1];

p[i1] = p[i1 + 1];

p[i1 + 1] = tmp;

}

}

}

int BinSearch(Student\* p, const int N, const string prizv, double MathG, double average)

{

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

double avg;

{

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

{

}

do {

m = (L + R) / 2;

avg = (p[m].fizyka + p[m].matematyka + p[m].informatyka) / 3.0;

if (p[m].prizv == prizv && avg == average && p[m].informatyka == MathG)

return m;

if ((avg < average)

||

(avg == average &&

p[m].matematyka < MathG)

||

(avg == average) &&

p[m].matematyka == MathG &&

p[m].prizv < prizv)

{

L = m + 1;

}

else

{

R = m - 1;

}

} while (L <= R);

return -1;

}

}

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

{

int\* I = new int[N]; // створили індексний масив

double avgIi, avgIj;

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

I[i] = i;

int i, j, value;

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

{

value = I[i];

avgIi = (p[I[i]].fizyka + p[I[i]].matematyka + p[I[i]].informatyka) / 3.0;

for (j = i - 1;

avgIj = ((p[I[j]].fizyka + p[I[j]].matematyka + p[I[j]].informatyka) / 3.0),

j >= 0 && ((avgIj > avgIi) ||

(avgIj == avgIi &&

p[I[j]].matematyka < p[value].matematyka) ||

(avgIj == avgIi &&

p[I[j]].matematyka == p[value].matematyka) &&

(p[I[j]].prizv > p[value].prizv));

j--)

{

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

}

I[j + 1] = value;

}

return I;

}

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

{

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

<< endl;

cout << "\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_| Бали |"

<< endl;

cout << "| № | Прізвище | Курс | Спеціальність | Фізика | Математика | Інформатика |"

<< endl;

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

<< endl;

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

{

cout << "| " << setw(3) << right << i + 1 << " ";

cout << "| " << setw(13) << left << p[I[i]].prizv

<< "| " << setw(4) << right << p[I[i]].kurs << " "

<< "| " << setw(11) << left << p[I[i]].specialnist

<< "| " << setw(4) << right << p[I[i]].fizyka << " "

<< "| " << setw(4) << right << p[I[i]].matematyka << " "

<< "| " << setw(4) << right << p[I[i]].informatyka << " |" << endl;

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

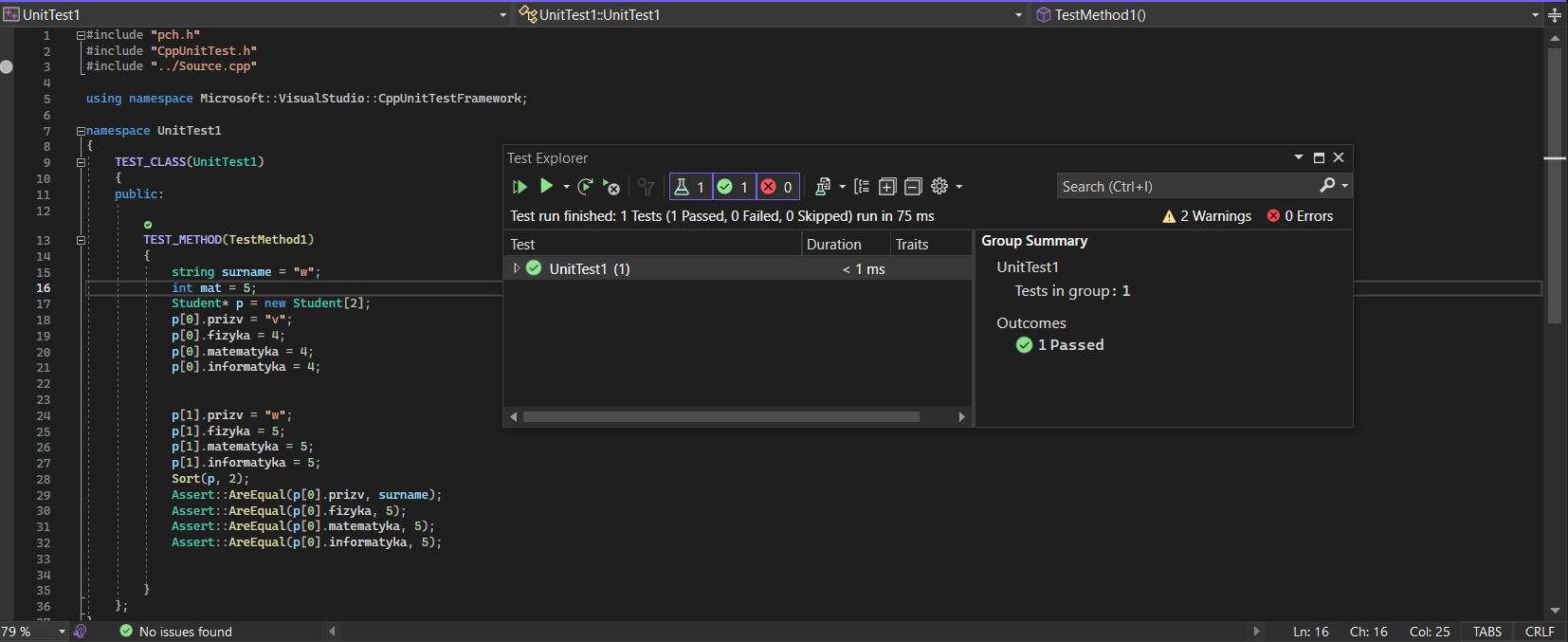
<< endl;

cout << endl;

}

}

**UNIT-test:**

****

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

<https://github.com/BigTrouble-Git/ashtray.git>

**Висновки**:

Виконавши цю лабораторну роботу я навчився впорядковувати масив структур з об’єднаннями. Навчився здійснювати фізичне та індексне впорядкування. Навчився здійснювати бінарний пошук у фізично чи індексно впорядкованому масиві.