**ЗВІТ**

**до лабораторної роботи № < 9.2.2 >**

**« Впорядкування та бінарний пошук в масиві структур »**

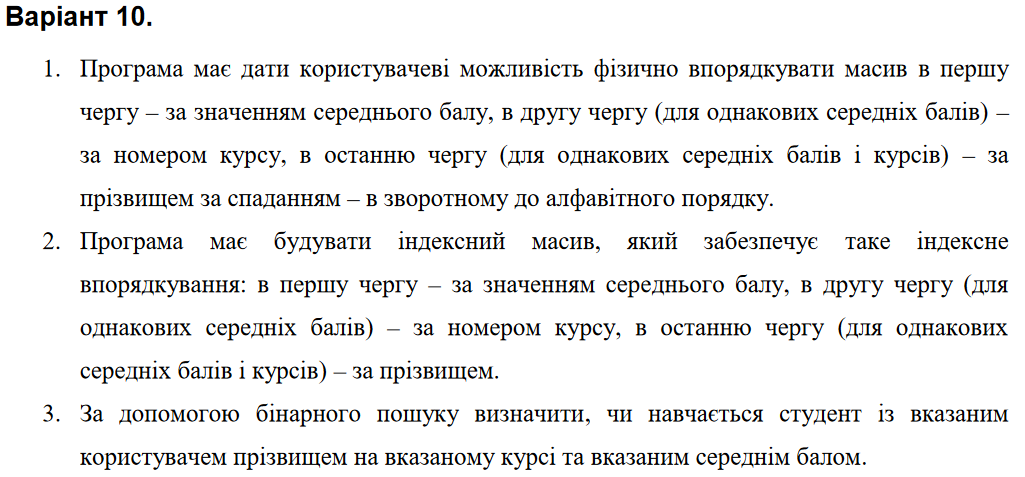
**з дисципліни**

**«Алгоритмізація та програмування»**

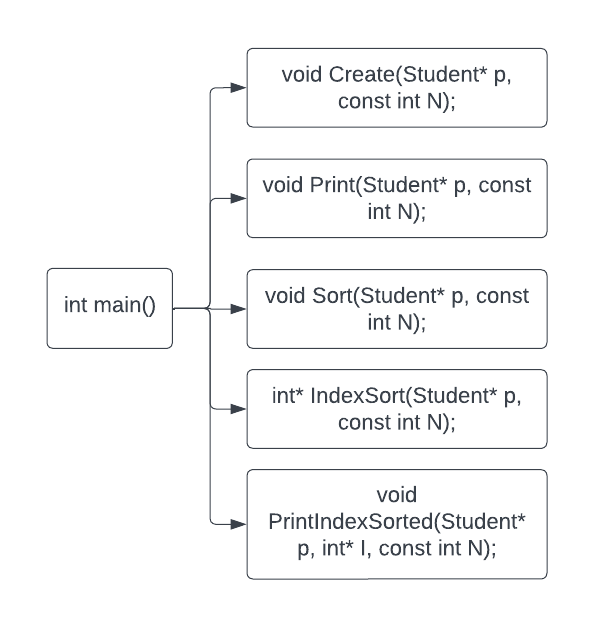
**студента групи ІН-105Б**

**Горанова Анастаса**

**Умова завдання:**



**Структурна схема:**



**Текст програми:**

// 9.2.2.cpp

// < Горанов Анастас >

// Лабораторна робота № 9.2.2

// Впорядкування та бінарний пошук в масиві структур

// Варіант 10

#include <iostream>

#include <iomanip>

#include <string>

using namespace std;

enum Specialnist { KN, ME, IN, FI, TN };

string specialnistList[] = { "KN", "ME", "IN", "FI", "TN" };

struct Student

{

string surname;

int course;

Specialnist spec;

int fiz;

int mat;

double average;

union

{

int prog;

int chis\_metod;

int pedagog;

};

};

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

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

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

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

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

int main()

{

int N;

cout << "Type the number of students: "; cin >> N;

Student\* p = new Student[N];

int menuItem;

do {

cout << endl << endl << endl;

cout << "Choose an action: " << endl << endl;

cout << " [1] - Data entry from the keyboard" << endl;

cout << " [2] - Outputting data to the screen" << endl;

cout << " [3] - Sorting" << endl;

cout << " [4] - Index sorting and its output" << endl;

cout << " [0] - Quit" << endl << endl;

cout << "Enter the value: "; 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);

cout << "Table was sorted!";

break;

case 4:

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

case 0:

break;

default:

cout << "Wrong value!" << endl;

}

} while (menuItem != 0);

return 0;

}

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

{

int spec;

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

{

cout << "Student # " << i + 1 << ":" << endl;

cin.get();

cin.sync();

cout << "Surname: "; getline(cin, p[i].surname);

cout << "Course: "; cin >> p[i].course;

cout << "Grades in physics: "; cin >> p[i].fiz;

cout << "Grades in math: "; cin >> p[i].mat;

cout << "A specialty (0 - KN, 1 - ME, 2 - IN, 3 - FI, 4 - TN): "; cin >> spec;

if (spec > 4) {

cout << "Wrong value, selected 2 (IN)" << endl;

spec = 2;

}

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

switch (p[i].spec)

{

case KN:

cout << "Grades in programming: "; cin >> p[i].prog;

p[i].average = (p[i].mat + p[i].fiz + p[i].prog) / 3.0;

break;

case ME:

cout << "Grades in pedagogy: "; cin >> p[i].pedagog;

p[i].average = (p[i].mat + p[i].fiz + p[i].pedagog) / 3.0;

break;

case IN:

cout << "Grades in num.methods : "; cin >> p[i].chis\_metod;

p[i].average = (p[i].mat + p[i].fiz + p[i].chis\_metod) / 3.0;

break;

case FI:

cout << "Grades in pedagogy: "; cin >> p[i].pedagog;

p[i].average = (p[i].mat + p[i].fiz + p[i].pedagog) / 3.0;

break;

case TN:

cout << "Grades in pedagogy: "; cin >> p[i].pedagog;

p[i].average = (p[i].mat + p[i].fiz + p[i].pedagog) / 3.0;

break;

}

cout << endl;

}

}

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

{

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

<< endl;

cout << "| # | Surname | Course | Specilization | Physic | Mathematic | Programming | Numerical Methods | Pedagogy |"

<< endl;

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

<< endl;

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

{

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

cout << "| " << setw(8) << left << p[i].surname

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

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

<< setw(4) << "| " << setw(3) << left << p[i].fiz << " "

<< setw(6) << "| " << setw(6) << left << p[i].mat << " ";

switch (p[i].spec)

{

case KN:

cout << setw(8) << "| " << setw(5) << left << p[i].prog << setw(21)

<< " | " << setw(10) << left

<< "| " << left << " |" << endl;

break;

case ME:

cout << "" << setw(13) << "| " << " "

<< "" << setw(20) << "| "

<< setw(6) << "| " << setw(4) << left << p[i].pedagog << " |" << endl;

break;

case IN:

cout << "" << setw(13) << "| " << setw(10) << " |" << " "

<< setw(9) << p[i].chis\_metod << " |" << " "

<< setw(8) << "" << " |" << endl;

break;

case FI:

cout << "" << setw(13) << "| " << " "

<< "" << setw(20) << "| "

<< setw(6) << "| " << setw(4) << left << p[i].pedagog << " |" << endl;

break;

case TN:

cout << "" << setw(13) << "| " << " "

<< "" << setw(20) << "| "

<< setw(6) << "| " << setw(4) << left << p[i].pedagog << " |" << endl;

break;

}

}

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

<< endl;

cout << endl;

}

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

{

Student tmp;

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

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

if ((p[i1].average < p[i1 + 1].average)

||

(p[i1].average = p[i1 + 1].average) &&

p[i1].course == p[i1 + 1].course

||

(p[i1].average = p[i1 + 1].average &&

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

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

{

tmp = p[i1];

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

p[i1 + 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[I[j]].average < p[value].average)

||

(p[I[j]].average == p[value].average &&

p[I[j]].course > p[value].course)

||

(p[I[j]].average == p[value].average &&

p[I[j]].course == p[value].course &&

p[I[j]].surname < p[value].surname));

j--)

{

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

}

I[j + 1] = value;

}

return I;

}

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

{

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

<< endl;

cout << "| # | Surname | Course | Specilization | Physic | Mathematic | Programming | Numerical Methods | Pedagogy |"

<< endl;

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

<< endl;

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

{

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

cout << "| " << setw(8) << left << p[I[i]].surname

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

<< " | " << setw(14) << left << specialnistList[p[I[i]].spec]

<< setw(4) << "| " << setw(3) << left << p[I[i]].fiz << " "

<< setw(6) << "| " << setw(6) << left << p[I[i]].mat << " ";

switch (p[I[i]].spec)

{

case KN:

cout << setw(8) << "| " << setw(5) << left << p[I[i]].prog << setw(21)

<< " | " << setw(10) << left

<< "| " << left << " |" << endl;

break;

case ME:

cout << "" << setw(13) << "| " << " "

<< "" << setw(20) << "| "

<< setw(6) << "| " << setw(4) << left << p[I[i]].pedagog << " |" << endl;

break;

case IN:

cout << "" << setw(13) << "| " << setw(10) << " |" << " "

<< setw(9) << p[I[i]].chis\_metod << " |" << " "

<< setw(8) << "" << " |" << endl;

break;

case FI:

cout << "" << setw(13) << "| " << " "

<< "" << setw(20) << "| "

<< setw(6) << "| " << setw(4) << left << p[I[i]].pedagog << " |" << endl;

break;

case TN:

cout << "" << setw(13) << "| " << " "

<< "" << setw(20) << "| "

<< setw(6) << "| " << setw(4) << left << p[I[i]].pedagog << " |" << endl;

break;

}

}

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

<< endl;

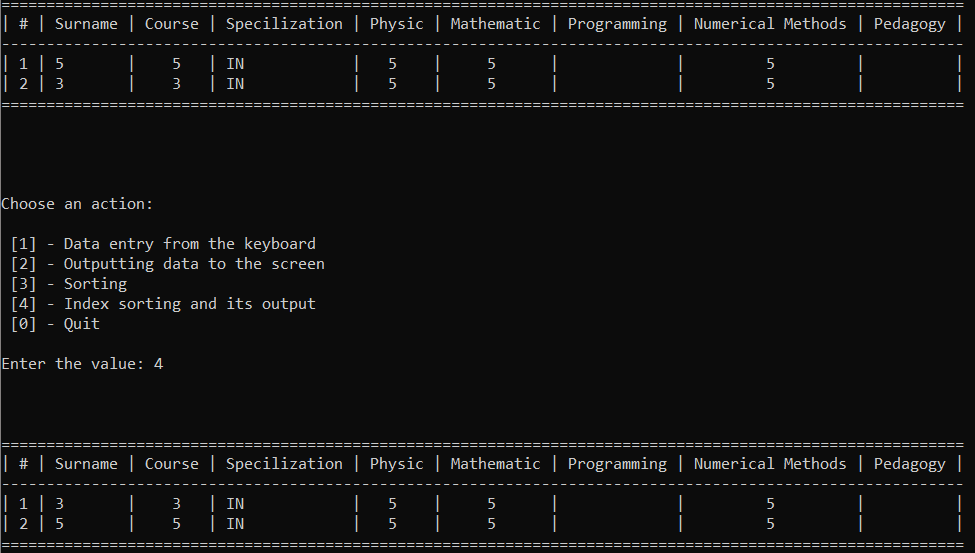
cout << endl;

}

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

*https://github.com/StassNG/9.2.2*

**Результати програми та unit-тесту:**

****

**Висновок:** я навчився впорядкуванню та бінарному пошуку в масиві структур.