*Національний університет харчових технологій*

*Інформаційних систем*

**Лабораторна робота №20**

з дисципліни *Основи програмування та алгоритмічні мови*

на тему: *Розв’язування задач з використанням класів*

Студент *1* курсу *2* групи

Спеціальності *122 «Комп’ютерні науки»*

*Держій Д.Ю*

Варіант завдання 4

Викладач *доцент к.т.н.*

*Грибков С.В*

Дата здачі *16.10.2022*

Оцінка

Київ-2022

Завдання 1:  
В першому класі реалізувати алгоритми по роботі з одновимірними масивами, що наведені в 6 лабораторній роботі. У методах класів реалізувати сортування матриць та векторів, що наведені в 11 лабораторній роботі.

Код програми:

#include <time.h>

#include <math.h>

#include <stdlib.h>

#include <iostream>

#include <stdio.h>

#include <windows.h>

#include <vector>

using namespace std;

class vec

{

public:

int n;

void rande();

void print();

int countp();

void element(int number);

void changeindex();

void sort();

private:

int e[100];

};

void vec::rande()

{

srand(time(NULL));

int i;

for (i = 0; i < n; i++)

e[i] = rand() % 100 - 50;

}

void vec::print()

{

int i;

printf("Сформований масив: \n", n);

for (i = 0; i < n; i++)

printf(" %d", e[i]);

cout << endl;

}

int vec::countp()

{

int count = 0;

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

{

if (e[i] > 0)

count++;

}

return count;

}

void vec::element(int number)

{

cout << "Елементи, менші за " << number << " та їх індекси: " << endl;

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

{

if (e[i] < number)

cout << "Елемент: " << e[i] << ", Індекс: " << i << endl;

}

}

void vec::changeindex()

{

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

{

if (e[i] % 2 == 0)

e[i] = i;

}

}

void vec::sort()

{

vector<int> positive;

int count = 0;

for (int i = 1; i < n; i += 2)

{

if (e[i] > 0)

{

positive.push\_back(e[i]);

count++;

}

}

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

{

for (int j = i + 1; j < count; j++)

{

if (positive[j] > positive[i])

{

int tmp = positive[i];

positive[i] = positive[j];

positive[j] = tmp;

}

}

}

int index = 1;

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

{

e[index] = positive[i];

index += 2;

}

}

int main()

{

system("cls");

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

vec r1;

cout << endl << "Введіть кількість елементів масиву = ";

cin >> r1.n;

r1.rande();

r1.print();

cout << "Кількість додатніх елементів: " << r1.countp() << endl;

int number;

cout << "Введіть число: ";

cin >> number;

r1.element(number);

r1.changeindex();

cout << "Масив після заміни парних елементів на їх індекси: " << endl;

r1.print();

r1.sort();

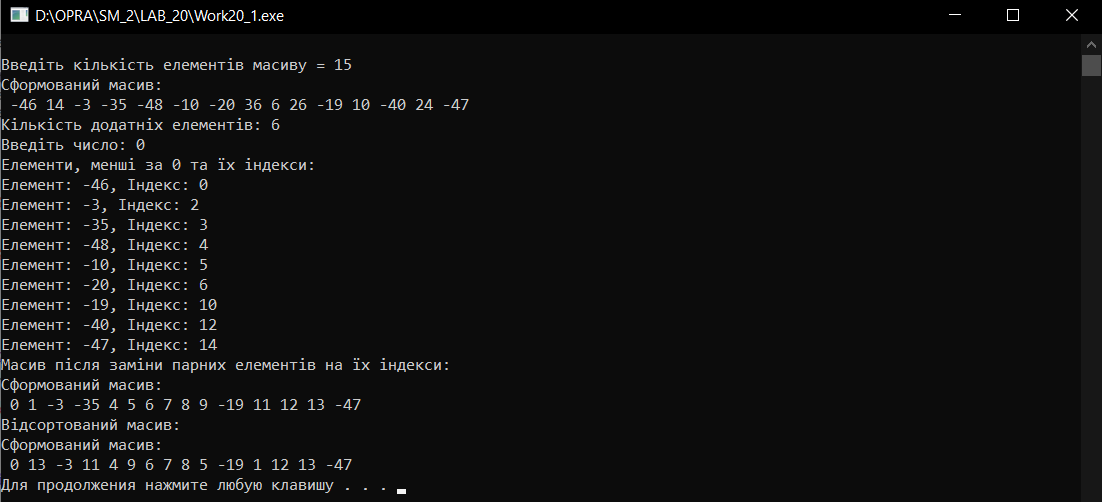
cout << "Відсортований масив: " << endl;

r1.print();

system("pause");

return 0;

}

Результат виконаної роботи  


Завдання 2:  
В першому класі реалізувати алгоритми по роботі з одновимірними масивами, що наведені в 6 лабораторній роботі. У методах класів реалізувати сортування матриць та векторів, що наведені в 11 лабораторній роботі.

Код програми:

#include <time.h>

#include <math.h>

#include <stdlib.h>

#include <iostream>

#include <stdio.h>

#include <windows.h>

#include <vector>

using namespace std;

class arr

{

public:

int rows;

int columns;

void rande();

void print();

int min();

int average();

vector<int> diagonal();

void sort();

private:

vector<vector<int> > arr;

};

void arr::rande()

{

srand(time(NULL));

arr.resize(rows);

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

{

arr[i].resize(columns);

for (int j = 0; j < columns; j++)

{

arr[i][j] = rand() % 100 - 50;

}

}

}

void arr::print()

{

printf("Матриця:\n", rows, columns);

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

{

for (int j = 0; j < columns; j++)

{

printf("%4d", arr[i][j]);

}

cout << endl;

}

}

int arr::min()

{

int min = arr[0][0];

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

{

for (int j = 0; j < columns; j++)

{

if (arr[i][j] < min)

{

min = arr[i][j];

}

}

}

return min;

}

int arr::average()

{

int sum = 0;

int count = 0;

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

{

for (int j = 0; j < columns; j++)

{

if (arr[i][j] > 0)

{

sum += arr[i][j];

count++;

}

}

}

if (count > 0)

{

return sum / count;

}

return 0;

}

vector<int> arr::diagonal()

{

vector<int> sums;

sums.resize(columns - 1);

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

{

for (int j = 0; j < columns; j++)

{

if (j > i && arr[i][j] > 0)

{

sums[j - 1] += arr[i][j];

}

}

}

return sums;

}

void arr::sort()

{

for (int j = 1; j < columns; j += 2)

{

int i = 1;

while (i < rows)

{

if (arr[i][j] < arr[i - 1][j])

{

swap(arr[i][j], arr[i - 1][j]);

if (i > 1)

{

i--;

}

}

else

{

i++;

}

}

}

}

int main()

{

system("cls");

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

arr r1;

cout << endl

<< "Введіть кількість рядків матриці: ";

cin >> r1.rows;

cout << "Введіть кількість стовпців матриці: ";

cin >> r1.columns;

r1.rande();

r1.print();

int min = r1.min();

cout << "Мінімальний елемент матриці: " << min << endl;

int average = r1.average();

cout << "Середнє арифметичне значення додатніх елементів матриці: " << average << endl;

vector<int> diagonal = r1.diagonal();

cout << "Масив діагоналей: ";

for (int i = 0; i < diagonal.size(); i++)

{

cout << diagonal[i] << " ";

}

cout << endl;

r1.sort();

cout << "Матриця після сортування непарних стовпців за зростанням:" << endl;

r1.print();

system("pause");

return 0;

}

Результат виконаної роботи  
