Міністерство освіти і науки України

Львівський національний університет імені Івана Франка

Факультет електроніки та комп’ютерних технологій

ЛАБОРАТОРНА РОБОТА № 2

з курсу “**Дискретна математика**”

“**Прості методи сортування**”

Виконав:

Ст. ФEI-13

Назаренко Олександр

Перевірив:

доцент Романишин Ростислав

Львів 2024

**Мета роботи:** Познайомитись з простими методами сортування: вибору, обміном, вставкою

**Постановка задачі:** Реалізувати методи сортування мовою C++.

**Хід роботи**

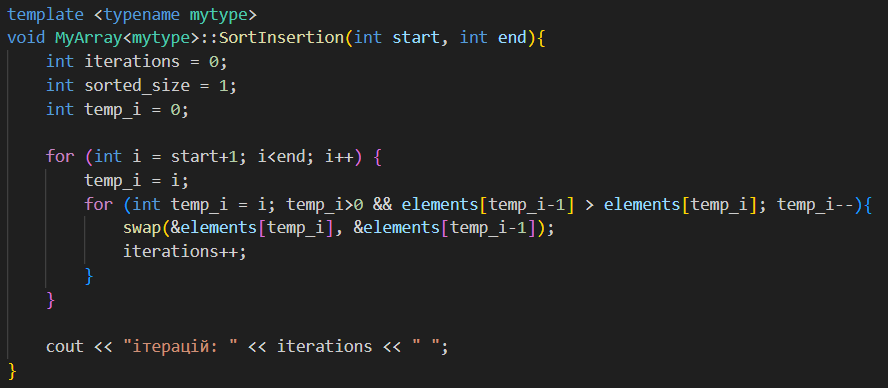
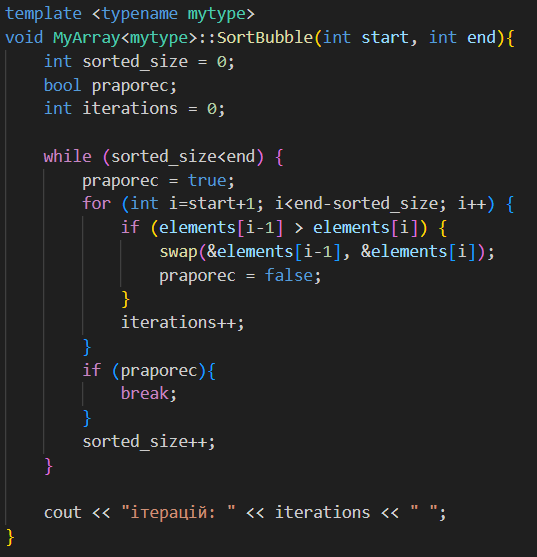
1. Створити нову бібліотеку Sort (файли Sort.h, Sort.cpp).

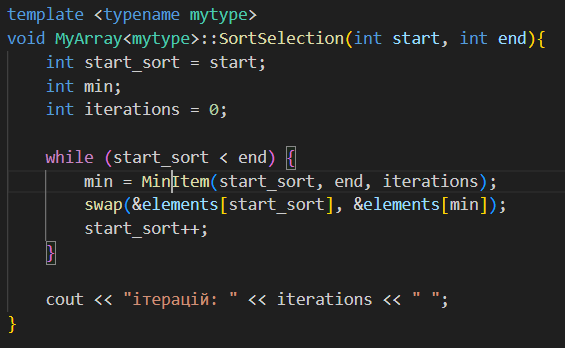


2. У файлі Sort.h за допомогою команди typedef зв’язати тип ключа елементу масиву datatype з типом даних, заданим викладачем.

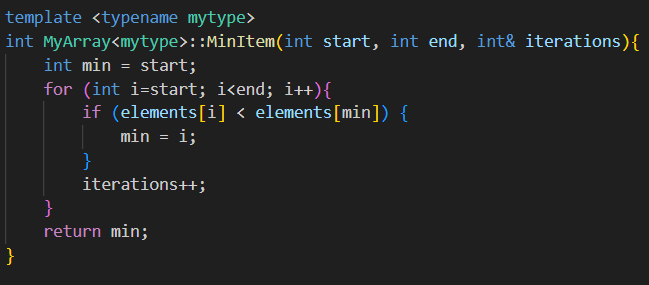
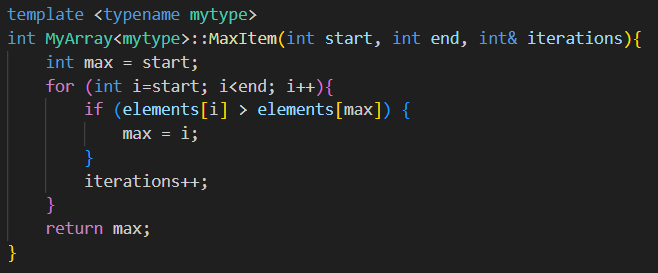
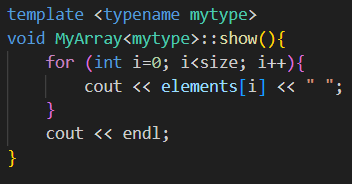
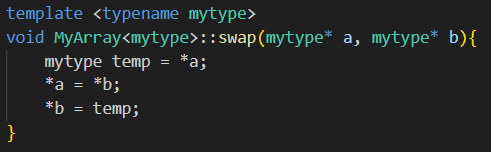


3. У бібліотеці Sort, згідно описаних вище алгоритмів, реалізувати функції сортування методом обміну (бульбашки), вставки та вибору (функції SortBubble(…), SortInsertion(…), SortSelection(…)).

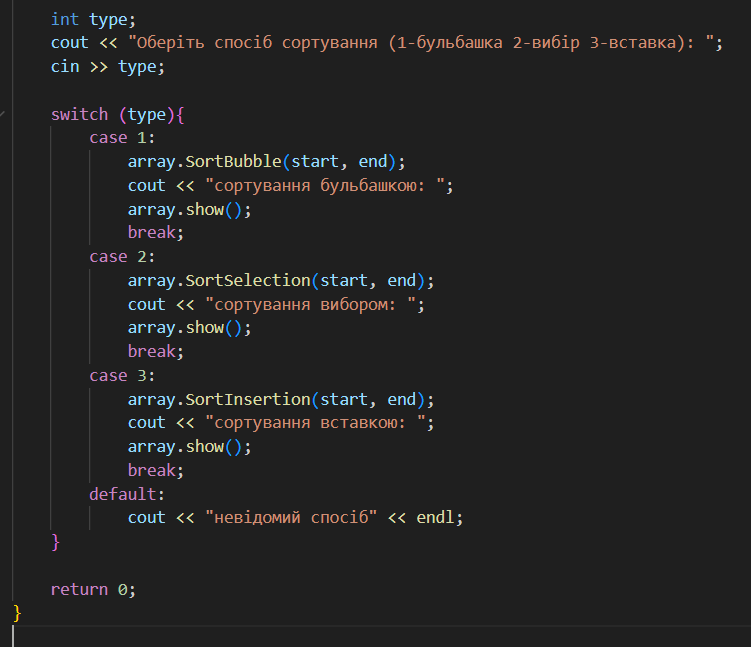




4. В цій же бібліотеці запрограмувати допоміжні функції для перестановки елементів масиву (swap(…)), відображення вмісту масиву (show(…)) та пошуку елементу з мінімальним та максимальним ключем у частині масиву (MinItem(…), MaxItem(…)).



5. Створити новий проект Lab\_2 до якого підключити бібліотеку Sort. У функції main() проекту реалізувати меню для вибору методу сортування масиву з даними.





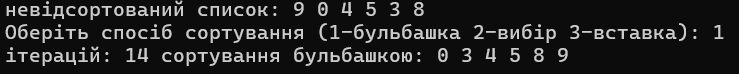
6. Модифікувати алгоритми сортування таким чином, щоб вони дозволяли відсортовувати тільки задану частину масиву.

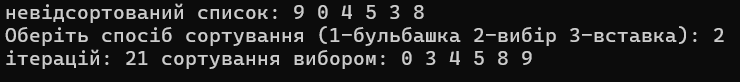


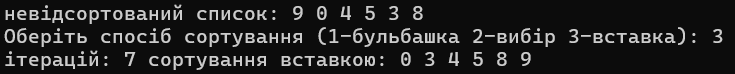
7. Отримати від викладача завдання: масив даних відповідного типу. Продемонструвати викладачеві результат сортування цього масиву різними методами та для різних ділянок масиву.



**Результати тестування**

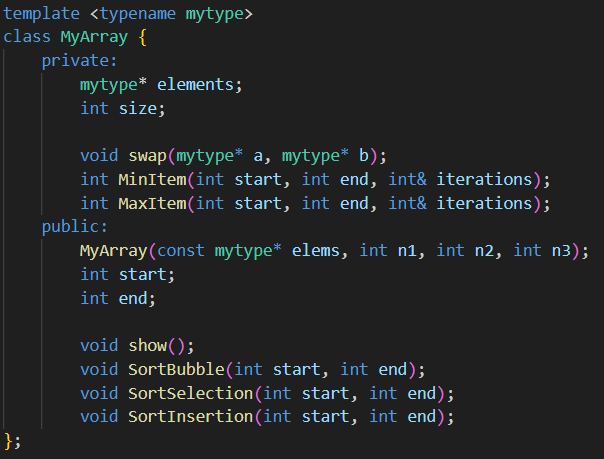






**Додаткове завдання**

Клас, шаблон.



**Висновок**

Методи сортування реалізовано. Функції правильно сортують масив. Після цієї лабораторної роботи я дізнався про прості методи сортування списку з чисел.

**КОД ПРОГРАМИ**

Lab2.cpp

#include <iostream>

#include "Sort.h"

#include <windows.h>

using namespace std;

int main() {

mytype items[] = {9, 0, 4, 5, 3, 8};

int size = sizeof(items) / sizeof(items[0]);

int start = 0;

int end = size;

MyArray<mytype> array(items, size, start, end);

//MyArray<mytype> array1(items, size, start, end);

//MyArray<mytype> array2(items, size, start, end);

SetConsoleOutputCP(CP\_UTF8); // вивід до консолі українських літер

cout << "невідсортований список: ";

array.show();

int type;

cout << "Оберіть спосіб сортування (1-бульбашка 2-вибір 3-вставка): ";

cin >> type;

switch (type){

case 1:

array.SortBubble(start, end);

cout << "сортування бульбашкою: ";

array.show();

break;

case 2:

array.SortSelection(start, end);

cout << "сортування вибором: ";

array.show();

break;

case 3:

array.SortInsertion(start, end);

cout << "сортування вставкою: ";

array.show();

break;

default:

cout << "невідомий спосіб" << endl;

}

return 0;

}

Sort.cpp

#include <iostream>

using namespace std;

template <typename mytype>

void MyArray<mytype>::swap(mytype\* a, mytype\* b){

mytype temp = \*a;

\*a = \*b;

\*b = temp;

}

template <typename mytype>

MyArray<mytype>::MyArray(const mytype\* elems, int n1, int n2, int n3) : size(n1), start(n2), end(n3) {

elements = new mytype[size];

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

elements[i] = elems[i];

}

}

template <typename mytype>

void MyArray<mytype>::show(){

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

cout << elements[i] << " ";

}

cout << endl;

}

template <typename mytype>

void MyArray<mytype>::SortBubble(int start, int end){

int sorted\_size = 0;

bool praporec;

int iterations = 0;

while (sorted\_size<end) {

praporec = true;

for (int i=start+1; i<end-sorted\_size; i++) {

if (elements[i-1] > elements[i]) {

swap(&elements[i-1], &elements[i]);

praporec = false;

}

iterations++;

}

if (praporec){

break;

}

sorted\_size++;

}

cout << "ітерацій: " << iterations << " ";

}

template <typename mytype>

int MyArray<mytype>::MaxItem(int start, int end, int& iterations){

int max = start;

for (int i=start; i<end; i++){

if (elements[i] > elements[max]) {

max = i;

}

iterations++;

}

return max;

}

template <typename mytype>

int MyArray<mytype>::MinItem(int start, int end, int& iterations){

int min = start;

for (int i=start; i<end; i++){

if (elements[i] < elements[min]) {

min = i;

}

iterations++;

}

return min;

}

template <typename mytype>

void MyArray<mytype>::SortSelection(int start, int end){

int start\_sort = start;

int min;

int iterations = 0;

while (start\_sort < end) {

min = MinItem(start\_sort, end, iterations);

swap(&elements[start\_sort], &elements[min]);

start\_sort++;

}

cout << "ітерацій: " << iterations << " ";

}

template <typename mytype>

void MyArray<mytype>::SortInsertion(int start, int end){

int iterations = 0;

int sorted\_size = 1;

int temp\_i = 0;

for (int i = start+1; i<end; i++) {

temp\_i = i;

for (int temp\_i = i; temp\_i>0 && elements[temp\_i-1] > elements[temp\_i]; temp\_i--){

swap(&elements[temp\_i], &elements[temp\_i-1]);

iterations++;

}

}

cout << "ітерацій: " << iterations << " ";

}

Sort.h

#include <iostream>

using namespace std;

typedef int mytype;

template <typename mytype>

class MyArray {

private:

mytype\* elements;

int size;

void swap(mytype\* a, mytype\* b);

int MinItem(int start, int end, int& iterations);

int MaxItem(int start, int end, int& iterations);

public:

MyArray(const mytype\* elems, int n1, int n2, int n3);

int start;

int end;

void show();

void SortBubble(int start, int end);

void SortSelection(int start, int end);

void SortInsertion(int start, int end);

};

#include "Sort.cpp"