Міністерство освіти і науки України

Національний університет «Львівська політехніка»

Кафедра ЕОМ



**Лабораторна робота № 1**

з дисципліни: «Адгоритми та методи обчислень»,

на тему: «Алгоритм. Властивості, параметри і характеристики складності алгоритму.»  
Варіант № 25

Виконав: ст. гр. КІ-28

Степанов В.В.

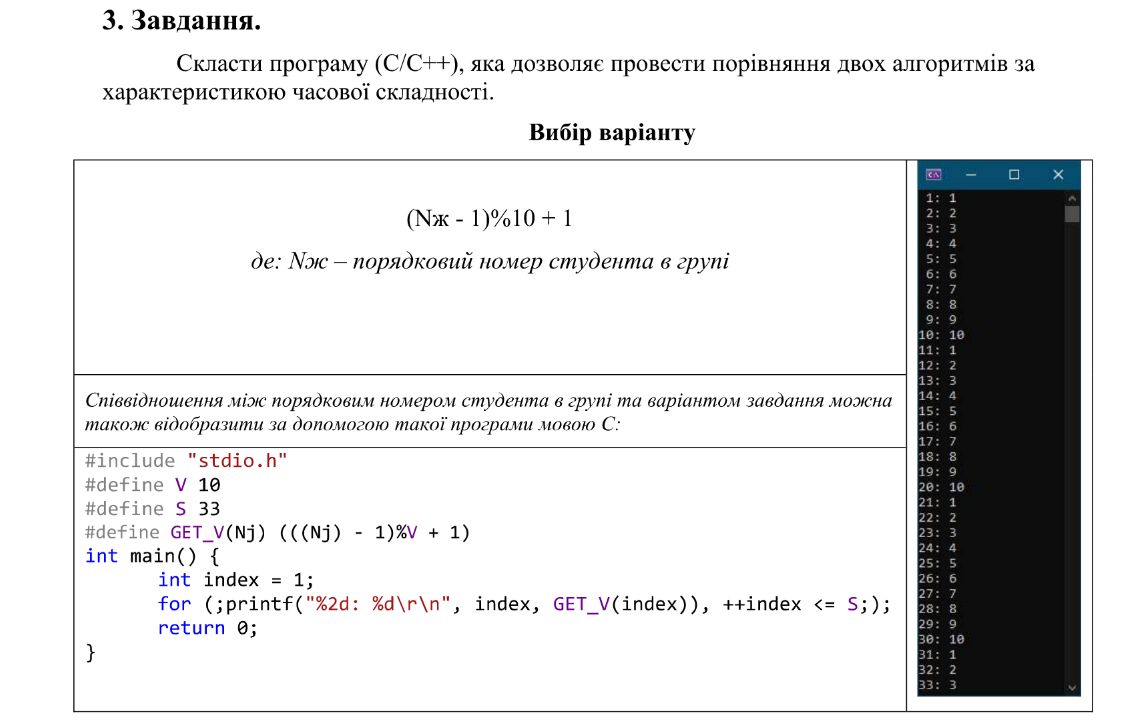
Прийняв:

Замроз П.І.

Львів – 2022

**Мета:** Проаналізувати складність алгоритму

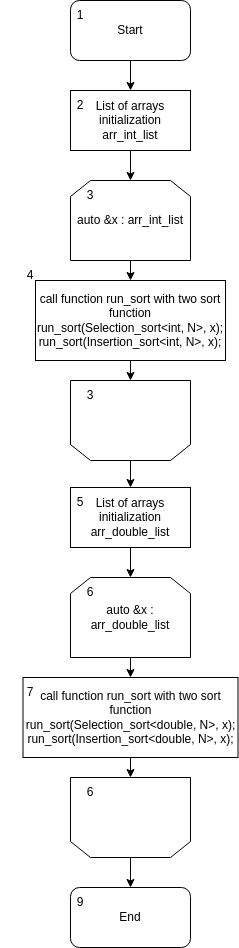
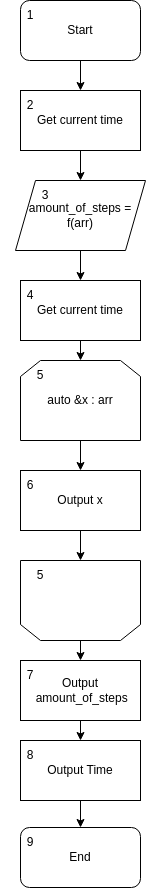
**ЗАВДАННЯ та дані згідно варіанту**



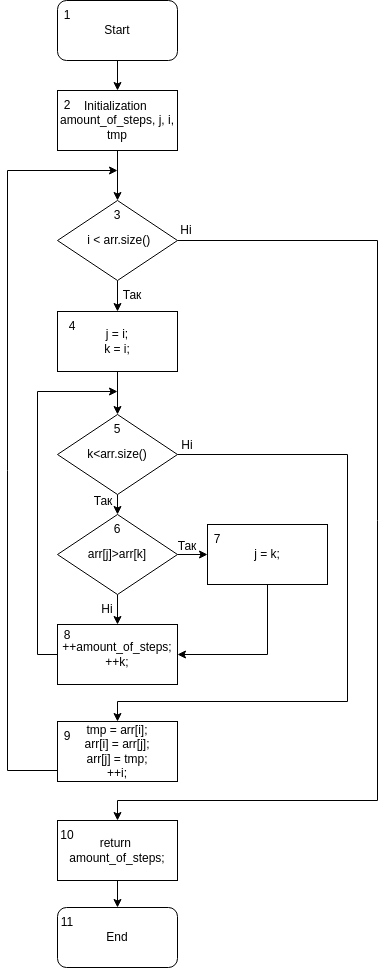
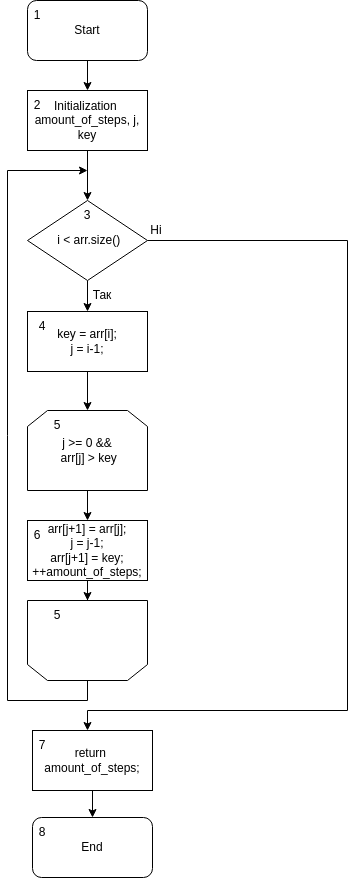
**РОЗВ’ЯЗУВАННЯ**

**Алгоритм програми:**

**Функція main(): Функція run\_sort():**

**** 

**Функція Selection\_sort(): Функція Insertion\_sort():**

**** 

**Код main.cpp:**

#include <iostream>

#include <array>

#include <vector>

#include <functional>

#include <chrono>

#define N 20

#define N\_DOUBLE 15

template <typename T, std::size\_t size>

std::size\_t Selection\_sort(std::array<T, size> &arr)

{

std::cout << "\t\t\t\t< Selection\_sort >" << std::endl;

std::size\_t amount\_of\_steps = 0;

std::size\_t j = 0;

T tmp;

for(std::size\_t i=0; i<arr.size(); i++)

{

j = i;

for(std::size\_t k = i; k<arr.size(); k++)

{

if(arr[j]>arr[k])

{

j = k;

}

++amount\_of\_steps;

}

tmp = arr[i];

arr[i] = arr[j];

arr[j] = tmp;

}

return amount\_of\_steps;

}

template <typename T, std::size\_t size>

std::size\_t Insertion\_sort(std::array<T, size> &arr)

{

std::cout << "\t\t\t\t< Insertion\_sort >" << std::endl;

std::size\_t amount\_of\_steps = 0;

int j = 0;

T key = 0;

for(std::size\_t i = 1;i<arr.size();i++){

key = arr[i];

j = i-1;

while(j >= 0 && arr[j] > key){

arr[j+1] = arr[j];

j = j-1;

arr[j+1] = key;

++amount\_of\_steps;

}

}

return amount\_of\_steps;

}

template <typename T, std::size\_t size>

void run\_sort(std::function<std::size\_t(std::array<T, size> &)> f, std::array<T, size> arr)

{

std::cout << "##############################################################################" << '\n';

auto t\_start = std::chrono::high\_resolution\_clock::now();

std::size\_t amount\_of\_steps = f(arr);

auto t\_end = std::chrono::high\_resolution\_clock::now();

for(auto &x : arr)

std::cout << x << ", ";

std::cout << std::endl;

std::cout << "Amount of steps: " << amount\_of\_steps << std::endl;

std::cout << "Time: " << std::chrono::duration<double, std::milli>(t\_end-t\_start).count() << std::endl;

std::cout << "###############################################################################" << "\n\n";

}

int main()

{

std::vector<std::array<int, N>> arr\_int\_list = {

{1, 4, 2, 8, 4, 5, 11, 2, 1, 1, 4, 3, 44, 22, -1, 2, 2, 11, 50 ,20},

{7, 5, 3, 2, 3, 2, 1, 5, 7, 9, 6, 5, 66, 1, 3, 9, 5, 3, 6, 7}};

for(auto &x : arr\_int\_list){

run\_sort(Selection\_sort<int, N>, x);

run\_sort(Insertion\_sort<int, N>, x);

}

std::cout <<"\n\n--------------------------\n\n\n\n";

std::vector<std::array<double, N>> arr\_double\_list = {

{1.6, 4.2, 2.1, 8.2, 4.6, 5.4, 11.11, 2.22, 1.3, 1.6, 4.5, 3.9, 44.0, 22.56, 1.11, 2.65, 0.001, 11.1, 50.1 ,20.2},

{7.8, 5.66, 3.35, 2.77, 3.12, 2.22, 1.55, 5.43, 7.77, 9.98, 6.34, 5.33, 66.77, 1.12, 3.45, 9.7, 5.034, 3.33, 6.1, 7.2}};

for(auto &x : arr\_double\_list){

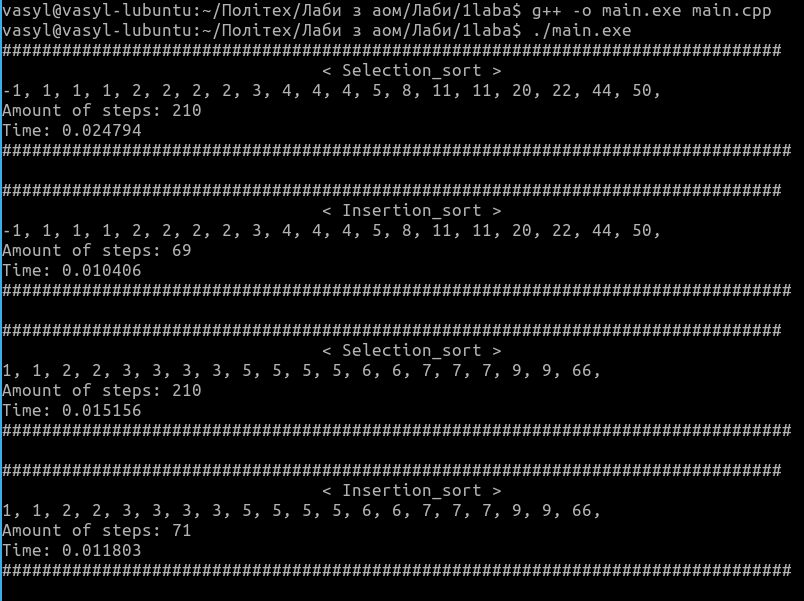
run\_sort(Selection\_sort<double, N>, x);

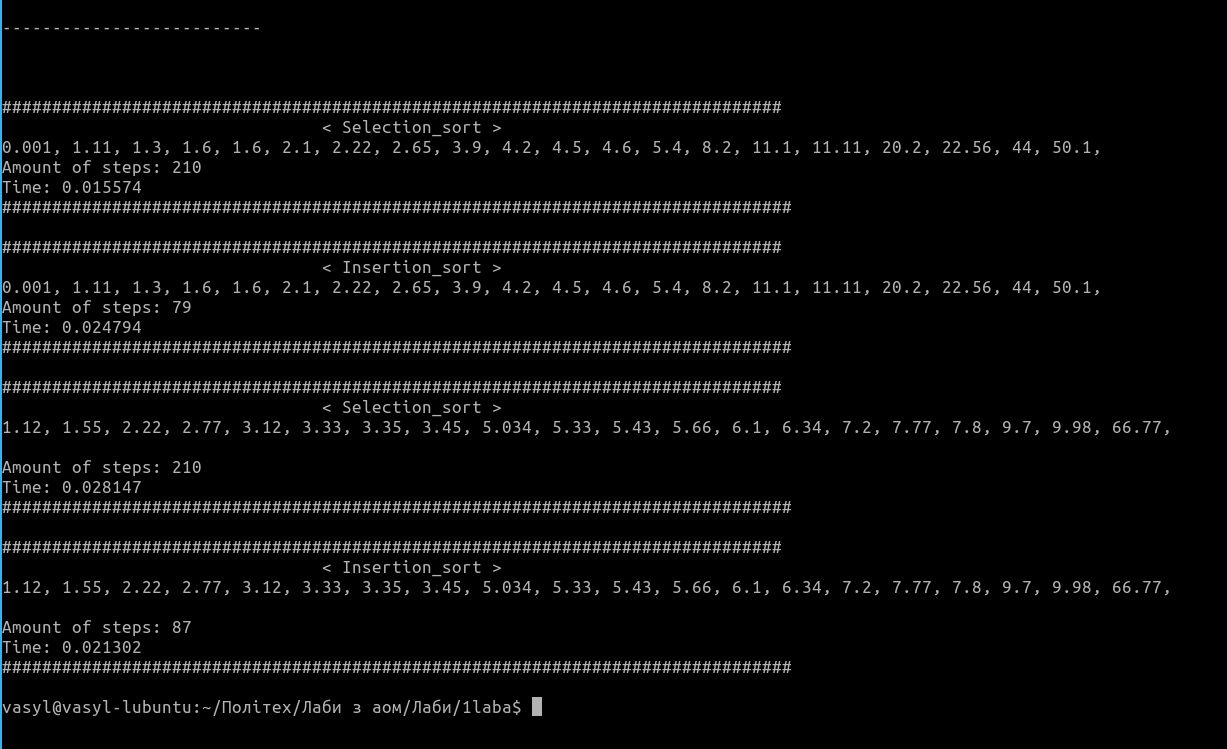
run\_sort(Insertion\_sort<double, N>, x);

}

return 0;}

**Скріншоти програми:**





**Висновок:** Я навчився аналізувати складність алгоритму