Любченко Олег 35 група

Завдання 1  
#include <iostream>

#include <chrono>

#include <Windows.h>

// Функція для лінійного пошуку

int linearSearch(int arr[], int size, int key, int &comparisons) {

    for (int i = 0; i < size; ++i) {

        comparisons++;

        if (arr[i] == key) {

            return i; // повертаємо індекс, якщо знайдено

        }

    }

    return -1; // якщо не знайдено

}

int main() {

    SetConsoleCP(65001);

    SetConsoleOutputCP(65001);

    const int maxSize = 50;

    int arr[maxSize];

    // Генерація масиву

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

        arr[i] = rand() % 256; // генеруємо випадкові числа від 0 до 255

    }

    // Виведення масиву на екран

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

        std::cout << arr[i] << " ";

    }

    std::cout << std::endl;

    int key;

    std::cout << "Введіть число для пошуку: ";

    std::cin >> key;

    int comparisons = 0; // кількість порівнянь

    auto start = std::chrono::high\_resolution\_clock::now(); // початок вимірювання часу

    int result = linearSearch(arr, maxSize, key, comparisons); // пошук

    auto end = std::chrono::high\_resolution\_clock::now(); // кінець вимірювання часу

    std::chrono::duration<double> duration = end - start; // обчислення тривалості

    if (result != -1) {

        std::cout << "Елемент знайдено на позиції: " << result << std::endl;

    } else {

        std::cout << "Елемент не знайдено." << std::endl;

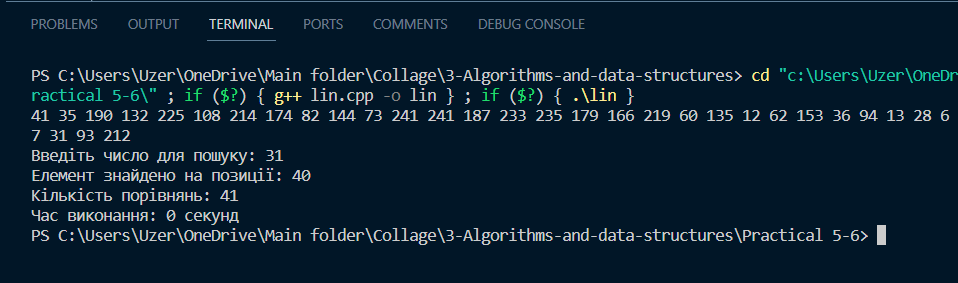
    }

    std::cout << "Кількість порівнянь: " << comparisons << std::endl;

    std::cout << "Час виконання: " << duration.count() << " секунд" << std::endl;

    return 0;

}



Завдання 2  
#include <iostream>

#include <chrono>

#include <algorithm>

#include <windows.h>

// Функція для бінарного пошуку

int binarySearch(int arr[], int size, int key, int &comparisons) {

    int left = 0;

    int right = size - 1;

    while (left <= right) {

        comparisons++;

        int mid = left + (right - left) / 2;

        if (arr[mid] == key) {

            return mid; // повертаємо індекс, якщо знайдено

        }

        if (arr[mid] < key) {

            left = mid + 1;

        } else {

            right = mid - 1;

        }

    }

    return -1; // якщо не знайдено

}

int main() {

    SetConsoleCP(65001);

    SetConsoleOutputCP(65001);

    const int maxSize = 50;

    int arr[maxSize];

    // Генерація масиву

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

        arr[i] = rand() % 256; // генеруємо випадкові числа від 0 до 255

    }

    std::sort(arr, arr + maxSize); // сортуємо масив для бінарного пошуку

    // Виведення масиву на екран

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

        std::cout << arr[i] << " ";

    }

    std::cout << std::endl;

    int key;

    std::cout << "Введіть число для пошуку: ";

    std::cin >> key;

    int comparisons = 0; // кількість порівнянь

    auto start = std::chrono::high\_resolution\_clock::now(); // початок вимірювання часу

    int result = binarySearch(arr, maxSize, key, comparisons); // пошук

    auto end = std::chrono::high\_resolution\_clock::now(); // кінець вимірювання часу

    std::chrono::duration<double> duration = end - start; // обчислення тривалості

    if (result != -1) {

        std::cout << "Елемент знайдено на позиції: " << result << std::endl;

    } else {

        std::cout << "Елемент не знайдено." << std::endl;

    }

    std::cout << "Кількість порівнянь: " << comparisons << std::endl;

    std::cout << "Час виконання: " << duration.count() << " секунд" << std::endl;

    return 0;

}

