Чернівецький національний університет імені Юрія Федьковича

Навчально-науковий інститут фізико-технічних та комп’ютерних наук

Кафедра математичних проблем управління і кібернетики

Звіт

про виконання лабораторної роботи No1

«Введення – виведення в С++. Файлові потоки С++. Масиви. Інтегровані середовища

розробки програм мовою С++. Робота з репозиторіями на github.com. Засоби

інтелектуального налагодження та тестування виконавчого коду»

з дисципліни

«Об’єктно-орієнтоване програмування»

Виконала:

студентка 141-А+Б групи

Жданюк А.С.

Перевірив: канд.фіз.-мат. наук,

доцент Лазорик В.В.

Оцінка:

Дата захисту:

Чернівці 2024

**Тема:** Введення – виведення в С++. Файлові потоки С++. Масиви. Інтегровані середовища розробки програм мовою С++. Робота з репозиторіями на github.com. Засоби інтелектуального налагодження та тестування виконавчого коду.

**Завдання лабораторної роботи:**

1. Зайти в свій обліковий запис на github.com. Зайти в github classroom.

2. Клонувати репозиторій: <https://classroom.github.com/a/sr7ArSl> - на робочий комп’ютер. В репозиторії в функції main() викликається демонстраційний приклад виконання подібної лабораторної роботи, який змінюєте на власний код. Функції прикладу можна використовувати (модифікувати та використовувати) для розв’язання задачі.

3. Лабораторна робота складається з задач. Для задач згідно варіанту написати:

a. Функції введення масивів з консолі та запис масивів в текстовий файл(текстові файли);

b. Функції введення розміру масиву(масивів) з консолі, створення масиву(масивів) з використанням датчика випадкових чисел та запис масиву(масивів) в бінарний файл(бінарний файли);

c. Функції розв’язання задач згідно варіанту з використанням динамічних масивів. Вхідні дані з читаються з файлів, результати записуються в нові файли та виводиться в консоль.

d. Функцію читає дані з файлів в контейнер (vector, valarray або array) та з контейнера дані вводиться в консоль.

4. В процесі написання функцій розв’язання задач лабораторної роботи періодично здійснювати синхронізацію з репозиторієм на github.com, з поясненням виконаної роботи ( git add ., git commit -m”коментар”, git push).

5. У функції main() створити меню яке б включало створені функції.

6. Оформити звіт про виконання лабораторної роботи, здійснити

порівняння методів роботи з масивами.

7. Звіт відправити в для оцінювання в [https://moodle.chnu.edu.ua](https://moodle.chnu.edu.ua/)

**Мої варіанти задач:**

1) Із одновимірного масиву А розміру N побудувати масив В із елементів, які більші числа 10.

2) Знайти номер останнього мінімального елемента серед елементів, менших Т1 і розташованих до першого елемента, більшого Т2.

**Хід лабораторної роботи:**

**Перше завдання**.



#include <iostream>

#include <vector>

std::vector<int> buildArray(const int A[], int N) {

std::vector<int> B;

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

if (A[i] > 10) {

B.push\_back(A[i]);

}

}

return B;

}

int main() {

const int N = 10; // Задайте розмір масиву A

int A[N] = {5, 12, 8, 15, 3, 10, 18, 7, 9, 11}; // Приклад масиву A

// Побудова масиву B

std::vector<int> B = buildArray(A, N);

// Виведення масиву B

std::cout << "Масив B з елементами більшими за 10: ";

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

std::cout << B[i] << " ";

}

std::cout << std::endl;

return 0;

}

**Друге завдання.**



#include <iostream>

#include <vector>

#include <limits> // для numeric\_limits

int findLastMinElementIndex(const std::vector<int>& arr, int T1, int T2) {

int lastMinIndex = -1; // Індекс останнього мінімального елемента, початково встановлено на -1

int minElement = std::numeric\_limits<int>::max(); // Початкове значення мінімального елемента, встановлене на максимальне можливе значення int

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

if (arr[i] > T2) {

break; // Завершуємо цикл, якщо знайшли перший елемент більший за T2

}

if (arr[i] < T1) {

// Якщо знайдено елемент менший за T1, оновлюємо індекс останнього мінімального елемента та його значення

lastMinIndex = i;

minElement = arr[i];

} else if (arr[i] == T1 && lastMinIndex == -1) {

// Якщо знайдено елемент, рівний T1, і це перший елемент рівний T1, оновлюємо індекс останнього мінімального елемента

lastMinIndex = i;

}

}

return lastMinIndex;

}

int main() {

std::vector<int> arr = {5, 3, 8, 2, 7, 1, 4, 6}; // Приклад вхідного масиву

int T1 = 4; // Параметр T1

int T2 = 6; // Параметр T2

int lastIndex = findLastMinElementIndex(arr, T1, T2);

if (lastIndex != -1) {

std::cout << "Номер останнього мінімального елемента: " << lastIndex << std::endl;

} else {

std::cout << "Мінімальний елемент не знайдено." << std::endl;

}

return 0;

}