Міністерство освіти і науки України

Національний університет «Львівська політехніка»

Інститут комп’ютерних технологій, автоматики та метрології

Кафедра «Комп’ютеризовані системи автоматики»



ЗВІТ

про виконання лабораторної роботи №10

«РОБОТА З МАСИВАМИ (RUST)»

із навчальної дисципліни: «Алгоритми та структури даних»

Варіант №11

Виконав:

студент групи ІР-24

Кіндрат Віктор Романович

Прийняв:

к.т.н., доцент

Сиротюк С. В.

Львів 2024

**Завдання:**

Масив має 36 дійсних елементів. Знайти суму квадратів тих його елементів, які

знаходяться між другим найбільшим від’ємним елементом і першим найменшим додатним елементом.

**Програмний код мовою Rust:**

fn main() {

let arr: [f64; 36] = [

-3.5, -2.1, -4.3, 0.1, 0.2, -1.5, 5.2, -6.7, 7.8, 3.4, -2.9, 8.2, -9.4, 4.3, -5.1, 6.6,

1.2, -3.2, 0.8, 2.1, -8.9, 7.0, 4.5, -3.1, 1.8, -4.0, 9.5, 2.8, -6.2, -1.1, 3.9, 2.7, 4.6,

0.3, -0.2, -7.1,

];

let mut largest\_negatives = vec![];

let mut smallest\_positives = vec![];

// Find all negative and positive elements

for &num in arr.iter() {

if num < 0.0 {

largest\_negatives.push(num);

} else if num > 0.0 {

smallest\_positives.push(num);

}

}

// Sort negatives in descending order and positives in ascending order

largest\_negatives.sort\_by(|a, b| b.partial\_cmp(a).unwrap());

smallest\_positives.sort\_by(|a, b| a.partial\_cmp(b).unwrap());

// Get the second largest negative and the first smallest positive

let second\_largest\_negative = largest\_negatives

.get(1)

.expect("Not enough negative elements");

let first\_smallest\_positive = smallest\_positives

.get(0)

.expect("Not enough positive elements");

// Determine the indices of these elements in the array

let neg\_index = arr

.iter()

.position(|&x| x == \*second\_largest\_negative)

.expect("Second largest negative not found");

let pos\_index = arr

.iter()

.position(|&x| x == \*first\_smallest\_positive)

.expect("First smallest positive not found");

// Find the range between these two indices

let (start, end) = if neg\_index < pos\_index {

(neg\_index + 1, pos\_index)

} else {

(pos\_index + 1, neg\_index)

};

// Calculate the sum of squares of elements in this range

let sum\_of\_squares: f64 = arr[start..end].iter().map(|&x| x.powi(2)).sum();

println!("Second largest negative: {}", second\_largest\_negative);

println!("First smallest positive: {}", first\_smallest\_positive);

println!(

"Sum of squares between indices {} and {}: {}",

start, end, sum\_of\_squares

);

}

**Результат виконання програми:**

Second largest negative: -1.1

First smallest positive: 0.1

Sum of squares between indices 4 and 29: 729.2600000000001

**Висновок:**

Під час виконання лабораторної роботи я засовоїв методи роботи із масивами у мові програмування Rust. Розробив програмний код згідно з індивідуальним завданням. Розроблений код компілюється без помилок та попереджень. Після запуску скомпільованого коду отримую очікуваний результат.