

Міністерство освіти і науки України
Національний університет «Львівська політехніка»
Кафедра систем штучного інтелекту



Звіт

про виконання

Лабораторних та практичних робіт № 4,5

з дисципліни: «Мови та парадигми програмування»

з розділу: «Прості структури даних. Одновимірні масиви. Двовимірні Масиви.
Алгоритми обробки.»

Виконав:

студент групи ІІІ-13

Літовінський Роман

Львів 2023

Тема роботи:

Прості структури даних. Одновимірні масиви. Двовимірні Масиви. Алгоритми обробки.

Мета роботи:

Task 1 - Theory Education Activities

Task 2 - Requirements management (understand tasks) and design activities (draw flow diagrams and estimate tasks 3-8)

Task 3 - Lab# programming: VNS Lab 4

Task 4 - Lab# programming: VNS Lab 5

Task 5 - Lab# programming: Algotester Lab 2

Task 6 - Lab# programming: Algotester Lab 3

Task 7 - Practice# programming: Class Practice Task

Task 8 - Result Documentation Report and Outcomes Placement Activities (Docs and Programs on GitHub)

Task 9 - Results Evaluation and Release

Теоретичні відомості:

1) Теоретичні відомості з переліком важливих тем:

- Тема №1: Прості структури даних
- Тема №2: Одновимірні масиви
- Тема №3: . Двовимірні Масиви.
- Тема №4: Алгоритми обробки.
-

2) Індивідуальний план опрацювання теорії:

- Тема №1:Прості структури даних
 - o Джерела Інформації
 - <https://acode.com.ua/urok-64-struktury/#toc-0>
 -
 - o Що опрацьовано:
 - оголошення та визначення структур, їхня ініціалізація
 - o Статус: Ознайомлений
 - o Початок опрацювання теми:30.11
 - o Звершення опрацювання теми: 01.12
- Тема №2:Одновимірні масиви
 - o Джерела Інформації:
 - o <https://www.youtube.com/watch?v=ULdbOaMBPYc>
 - o

- Що опрацьовано:
 - Визначення масиву. Одновимірні масиви. Ініціалізація масиву
- Статус: Ознайомлений
- Початок опрацювання теми: 03.12
- Звершення опрацювання теми: 04.12
- Тема №3:.. Двовимірні Масиви.
 - Джерела Інформації:
 - <https://acode.com.ua/urok-81-bagatovymirni-masyvy/>
 - Що опрацьовано:
 - Двовимірні масиви. Масиви рядків. Багатовимірні масиви
 - Статус: Ознайомлений
 - Початок опрацювання теми: 03.12
 - Звершення опрацювання теми: 04.12

Виконання роботи:

1. Опрацювання завдання та вимог до програм та середовища:

Lab 2v2

Обмеження: 1 сек., 256 MiB

У вас є масив r розміром N . Також вам дано 3 цілих числа.

Спочатку ви маєте видалити з масиву ці 3 числа, які вам дані. Після цього перетворити цей масив у масив сум, розміром $N_{new} - 1$ (розмір нового масиву після видалення елементів), який буде відображати суми сусідніх елементів нового масиву.

Далі необхідно вивести масив сум на екран.

Вхідні дані

У першому рядку ціле число N - кількість чисел

У другому рядку масив r , який складається з N цілих чисел

У третьому рядку 3 цілих числа, a, b, c , які треба видалити з масиву

Вихідні дані

У першому рядку ціле число M - кількість чисел у масиві, який буде виведено

У наступному рядку M чисел - новий масив

Обмеження

$$1 \leq N \leq 10^5$$

$$0 \leq r_i \leq 10^5$$

$$0 \leq a, b, c \leq 10^5$$

Завдання №2 Algotester Lab 3

- Варіант завдання 2

Вам дано 2 масиви розміром N та M . Значення у цих масивах унікальні.

Ваше завдання вивести у першому рядку кількість елементів, які наявні в обох масивах одночасно, у другому кількість унікальних елементів в обох масивах разом.

Вхідні дані

У першому рядку ціле число N

у другому рядку N цілих чисел $a_1 \dots a_n$

У третьому рядку ціле число M

у четвертому рядку M цілих чисел $b_1 \dots b_n$

Вихідні дані

У першому рядку одне ціле число - кількість елементів, які наявні в обох масивах одночасно.

- У другому рядку кількість унікальних елементів в обох масивах (тобто кількість унікальних елементів у масиві, який буде об'єднанням двох даних).
-

Завдання №3 Перевірка чи слово або число є паліндромом

-

Задача

Реалізувати програму, яка перевіряє, чи дане слово чи число є паліндромом за допомогою рекурсії.

Паліндром — це послідовність символів, яка однаково читається вперед і назад (наприклад, «radar», «level», «12321»).

Мета Задачі

Навчитися користуватися механізмами перевантаження функції та використовувати рекурсію для вирішення задач обчислення.

Вимоги:

1. Визначення функції:
 - а. Реалізуйте рекурсивну функцію *isPalindrome*, яка перевіряє, чи заданий рядок є паліндромом.
2. Приклад визначення функції:
 - а. *bool isPalindrome(const string& str, int start, int end);*
3. Перевантаження функцій:
 - а. Перевантажте функцію *isPalindrome* для роботи з цілими значеннями.
 - б. *bool isPalindrome(ціле число);*
4. Рекурсія:
 - а. Рекурсивна функція для рядків перевірить символи в поточній початковій і кінцевій позиціях. Якщо вони збігаються, він буде рекурсивно перевіряти наступні позиції, поки початок не перевищить кінець, після чого рядок буде визначено як паліндром.

Кроки реалізації

- Визначте та реалізуйте рекурсивну функцію isPalindrome для рядків.
- Визначте та реалізуйте перевантажену функцію isPalindrome для цілих чисел. Перетворити ціле число на рядок і використовуйте рядкову версію функції, щоб перевірити, чи це паліндром.

Завдання №4 VNS Lab 4

варіант завдання 5

- 1) Сформувати одновимірний масив цілих чисел, використовуючи генератор випадкових чисел.
- 2) Роздрукувати отриманий масив.
- 3) Знищити елементи кратні 7.
- 4) Додати після кожного непарного елемента масиву елемент зі значенням 0.
- 5) Роздрукувати отриманий масив.

Завдання №5 VNS Lab 5

варіант завдання 5

Написати функцію для знищення рядка із двовимірною масиву. Рядки, що залишились, повинні бути розташовані щільно, елементи яких бракує замінюються 0. За допомогою розроблених функцій знищити з масиву рядки з номерами від A до B.

2. Дизайн та планована оцінка часу виконання завдань:

Програма №1 Algotester Lab 2

- Блок-схема

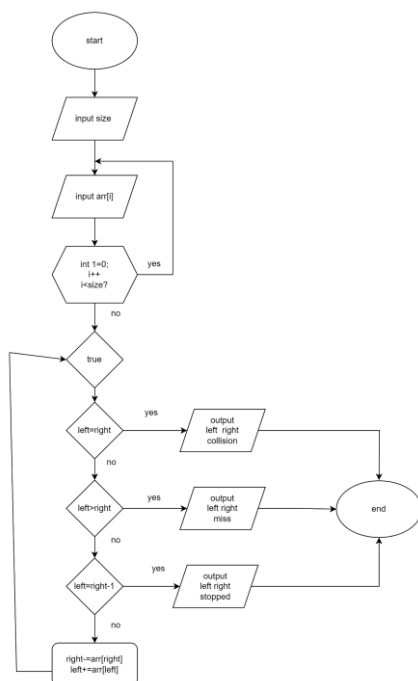


Figure 1.

- Планований час на реалізацію пів години
- Важливі деталі для врахування в імплементації

3. Код програм з посиланням на зовнішні ресурси:

Завдання №1 Algotester Lab 2

```
#include <iostream>
#include <vector>
using namespace std;
int main() {
    int N, m;
    cin >> N;
    vector<int> r(N);
    int a, b, c;
    cin >> a >> b >> c;
    for (int i = 0; i < N; i++) {
        cin >> r[i];
    }
    for (int i = 0; i < N; i++) {
        if (r[i] == a || r[i] == b || r[i] == c) {
            r.erase(r.begin() + i);
            i--;
            N--;
        }
    }
    m=N-1;
    if(N<2){
        m=0;
    }
    cout << m << endl;
    for (int i = 0; i < m; i++) {
        cout << r[i] + r[i + 1] << endl;
    }
    return 0;
}
```

Завдання №2 Algotester Lab 3

```
#include <iostream>
#include <unordered_set>

int main() {
    int N, M;
    std::cin >> N >> M;

    std::unordered_set<int> setN, setM;

    // Зчитування елементів першого масиву та занесення їх у множину setN
```

```

    for (int i = 0; i < N; ++i) {
        int num;
        std::cin >> num;
        setN.insert(num);
    }

    // Зчитування елементів другого масиву та занесення їх у множину setM
    for (int i = 0; i < M; ++i) {
        int num;
        std::cin >> num;
        setM.insert(num);
    }

    int commonElements = 0;
    // Підрахунок кількості елементів, які присутні в обох масивах одночасно
    for (int num : setN) {
        if (setM.find(num) != setM.end()) {
            commonElements++;
        }
    }

    // Кількість унікальних елементів у обох масивах разом
    int uniqueElements = setN.size() + setM.size() - commonElements;

    // Виведення результату
    std::cout << commonElements << std::endl;
    std::cout << uniqueElements << std::endl;

    return 0;
}

```

Завдання №3 Class Practice Task

```

#include <iostream>
#include <cstring>
#include <string>
using namespace std;

bool isPalindrome(const string& str) {
    for (int i = 0; i < str.length() / 2; i++) {
        if (str[i] != str[str.length() - i - 1]) {
            return false;
        }
    }
    return true;
}

```

```

}

bool isPalindrome(const char* str) {
    int length = strlen(str);
    for (int i = 0; i < length / 2; i++) {
        if (str[i] != str[length - i - 1]) {
            return false;
        }
    }
    return true;
}

int main() {
    string input_str;
    cout << "Enter a string: ";
    cin >> input_str;

    const char* input_cstr = input_str.c_str();

    if (isPalindrome(input_str)) {
        cout << "String is a palindrome." << endl;
    } else {
        cout << "String is not a palindrome." << endl;
    }

    return 0;
}

```

Завдання №4 VNS Lab 4

```

#include <iostream>
#include <cstdlib> // для використання srand() та rand()
#include <ctime> // для використання time() для ініціалізації генератора випадкових чисел

void printArray(int arr[], int size) {
    for (int i = 0; i < size; ++i) {
        std::cout << arr[i] << " ";
    }
    std::cout << std::endl;
}

int main() {
    const int arraySize = 10; // Розмір масиву
    int myArray[arraySize];

    // Ініціалізація генератора випадкових чисел
    std::srand(static_cast<unsigned int>(std::time(nullptr)));
}

```



```

    // Заповнення масиву випадковими числами
    for (int i = 0; i < arraySize; ++i) {
        myArray[i] = std::rand() % 101; // Генеруємо випадкові числа в діапазоні від
0 до 100
    }

    std::cout << "Початковий масив: ";
    printArray(myArray, arraySize);

    // Знищення останнього елемента, який дорівнює 0
    for (int i = arraySize - 1; i >= 0; --i) {
        if (myArray[i] == 0) {
            myArray[i] = 0;
            break;
        }
    }

    // Додавання після елемента масиву з заданим індексом елемента зі значенням 100
    int insertIndex = 3; // Заданий індекс для вставки
    if (insertIndex >= 0 && insertIndex < arraySize) {
        for (int i = arraySize - 1; i > insertIndex; --i) {
            myArray[i] = myArray[i - 1];
        }
        myArray[insertIndex + 1] = 100;
    }

    std::cout << "Масив після змін: ";
    printArray(myArray, arraySize);

    return 0;
}

```

Завдання №5 VNS Lab 5

```

#include <iostream>
#include <unordered_set>

int main() {
    int N, M;
    std::cin >> N >> M;

    std::unordered_set<int> setN, setM;

    // Зчитування елементів першого масиву та занесення їх у множину setN

```

```

for (int i = 0; i < N; ++i) {
    int num;
    std::cin >> num;
    setN.insert(num);
}

// Зчитування елементів другого масиву та занесення їх у множину setM
for (int i = 0; i < M; ++i) {
    int num;
    std::cin >> num;
    setM.insert(num);
}

int commonElements = 0;
// Підрахунок кількості елементів, які присутні в обох масивах одночасно
for (int num : setN) {
    if (setM.find(num) != setM.end()) {
        commonElements++;
    }
}

// Кількість унікальних елементів у обох масивах разом
int uniqueElements = setN.size() + setM.size() - commonElements;

// Виведення результату
std::cout << commonElements << std::endl;
std::cout << uniqueElements << std::endl;

return 0;
}

```

4. Результати виконання завдань, тестування та фактично затрачений час:

Завдання №1 Algotester Lab 2

```

PS D:\lpnu\Epic\epic 4 finish\output> & .\'lab4.exe'
initial array: 53 63 68 26 66 95 88 2 5 4
Array after changes: 53 63 68 26 100 66 95 88 2 5

```

Завдання №2 Algotester Lab 3

```

PS C:\Users\litov> cd 'd:\lpnu\Epic\epic 4 finish\output'
PS D:\lpnu\Epic\epic 4 finish\output> & .\'Algotester lab3.exe'

1
2
2
2
2
1
1

```

Завдання №3 Class Practice

```

PS D:\lpnu\Epic\epic 4 finish\output> cd 'd:\lpnu\Epic\epic 4 finish\output'
PS D:\lpnu\Epic\epic 4 finish\output> & .\'Class Practice Work.exe'
Enter a number or word to check if it's a palindrome: road
It's not a palindrome

```

Завдання №5 VNS Lab 5

```

PS D:\lpnu\Epic\epic 4 finish\output> & .\'lab5.exe'
1 2 3
4 5 6
1 7 8

```

Завдання №6 VNS Lab 4

```

PS D:\lpnu\Epic\epic 4 finish\output> & .\'lab4.exe'
initial array: 54 44 47 50 74 12 41 50 83 16
Array after changes: 54 44 47 50 100 74 12 41 50 83

```

Висновки:

В ході виконання лабораторних та практичних робіт з теми "Прості структури даних. Одновимірні масиви. Двовимірні Масиви. Алгоритми обробки" я реалізував на практиці свої теоретичні знання про прості структури даних, одновимірні масиви, двовимірні масиви, алгоритми обробки. Створив програму програму в алготестері яка шукає спільні та унікальні значення в двох масивах. Навчився працювати з різними типами масивів.