Міністерство освіти і науки України

Національний університет «Львівська політехніка»

Кафедра систем штучного інтелекту



**Звіт**

про виконання

**Лабораторних та практичних робіт №4**

***з дисципліни:*** «Мови та парадигми програмування»

***з розділу***: «Одновимірні масиви. Двовимірні Масиви. Алгоритми обробки»

***Виконала:***

студентка групи ШІ-14

Михальчук Віра Іванівна

# **Тема роботи:**

Ознайомлення з одновимірними та двовимірними масивами. Знайомство з базовими алгоритми обробки даних

# **Мета роботи:**

Ознайомитися з одновимірними та двовимірними у С++. Опрацювати та застосувати алгоритми обробки для масивів.

# **Теоретичні відомості:**

1. Теоретичні відомості з переліком важливих тем:

* **Тема №1:** Одновимірні масиви.
* **Тема №2:** Двовимірні масиви.
* **Тема №3:** Алгоритми обробки.
* **Тема №4:** Cтек.

1. Індивідуальний план опрацювання теорії:

* Тема №1: Одновимірні масиви.
  + Джерела Інформації
    - Відео. https://youtu.be/ULdbOaMBPYc?si=QqKuHITS\_eaT5Wxu
    - Стаття. <https://acode.com.ua/urok-77-masyvy/>
  + Що опрацьовано:
    - Переглянула відео де пояснюється синтаксис та функціонал одновимірних масивів.
    - Прочитавши статтю, закріпила нову інформацію.
  + Статус: Ознайомлена
  + Початок опрацювання теми: 04.12.23
  + Звершення опрацювання теми: 14.12.23
* Тема №2: Двовимірні масиви.
  + Джерела Інформації:
    - Відео. https://youtu.be/V2g3B9Zbh4Q?si=3LKgzcFKRfb3Rxqj
    - Стаття. <https://acode.com.ua/urok-81-bagatovymirni-masyvy/>
  + Що опрацьовано:
    - Переглянула відео.
    - Прочитала статтю.
  + Статус: Ознайомлена
  + Початок опрацювання теми: 04.12.23
  + Звершення опрацювання теми: 14.12.23
* Тема №3: Алгоритми обробки.
  + Джерела Інформації:
    - Відео. https://youtu.be/gncUL57AHwk?t=487

https://youtu.be/mSFZtI8ui4g?si=mnAE0LWOYgYsdYvo

* + - Стаття. <https://acode.com.ua/urok-80-sortuvannya-masyviv-metodom-vyboru/#toc-3>

<http://cpp.dp.ua/sortuvannya-masyviv/>

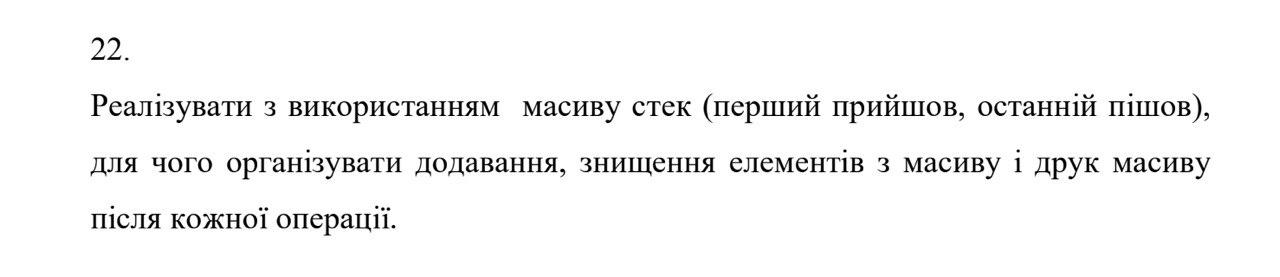
* + Що опрацьовано:
    - Переглянула відео про сортування вибіркою та швидке сортування
    - Прочитала статтю про сортування вибіркою та бульбашкою. Дізналась як позначати сортування на блок схемах.
  + Статус: Ознайомлена
  + Початок опрацювання теми: 04.12.23
  + Звершення опрацювання теми: 14.12.23
* Тема №4:Стек.
  + Джерела Інформації:
    - Відео. <https://youtu.be/WEwD-ZuTc1w?si=ooEIXZvLRz6QmAmf>
    - Стаття. <https://acode.com.ua/urok-111-stek-i-kupa/>
  + Що опрацьовано:
    - Переглянула відео.
    - Прочитала статтю.
  + Статус: Ознайомлена
  + Початок опрацювання теми: 04.12.23
  + Звершення опрацювання теми: 14.12.23

# **Виконання роботи:**

## **1. Опрацювання завдання та вимог до програм та середовища:**

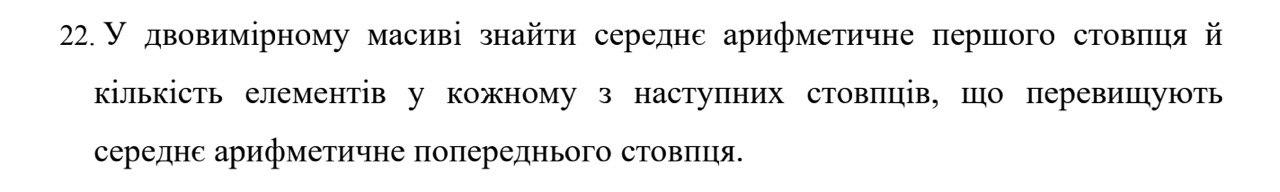
Завдання №1 VNS Lab 4

* Варіант 22
* Деталі завдання



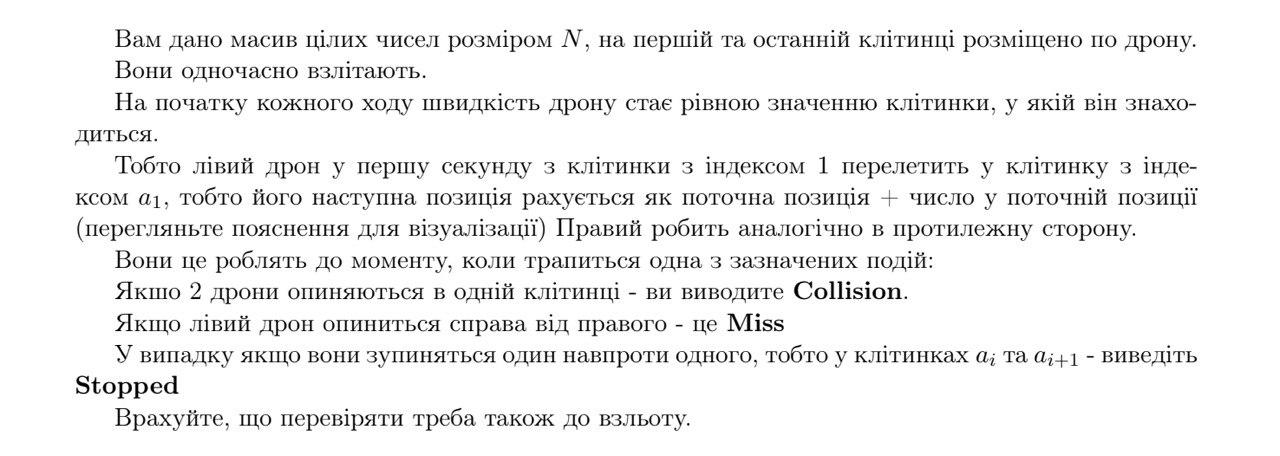
Завдання №2 VNS Lab 5

* Варіант 22
* Деталі завдання



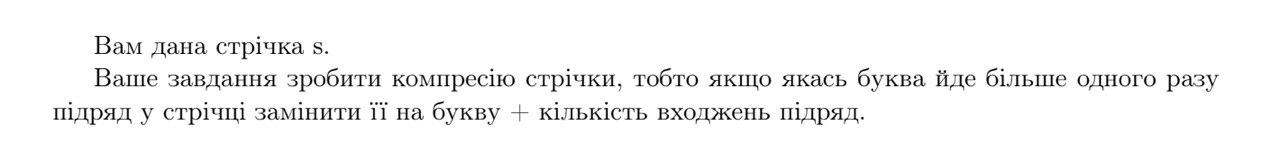
Завдання №3 Algotester Lab 2

* Варіант 3
* Деталі завдання



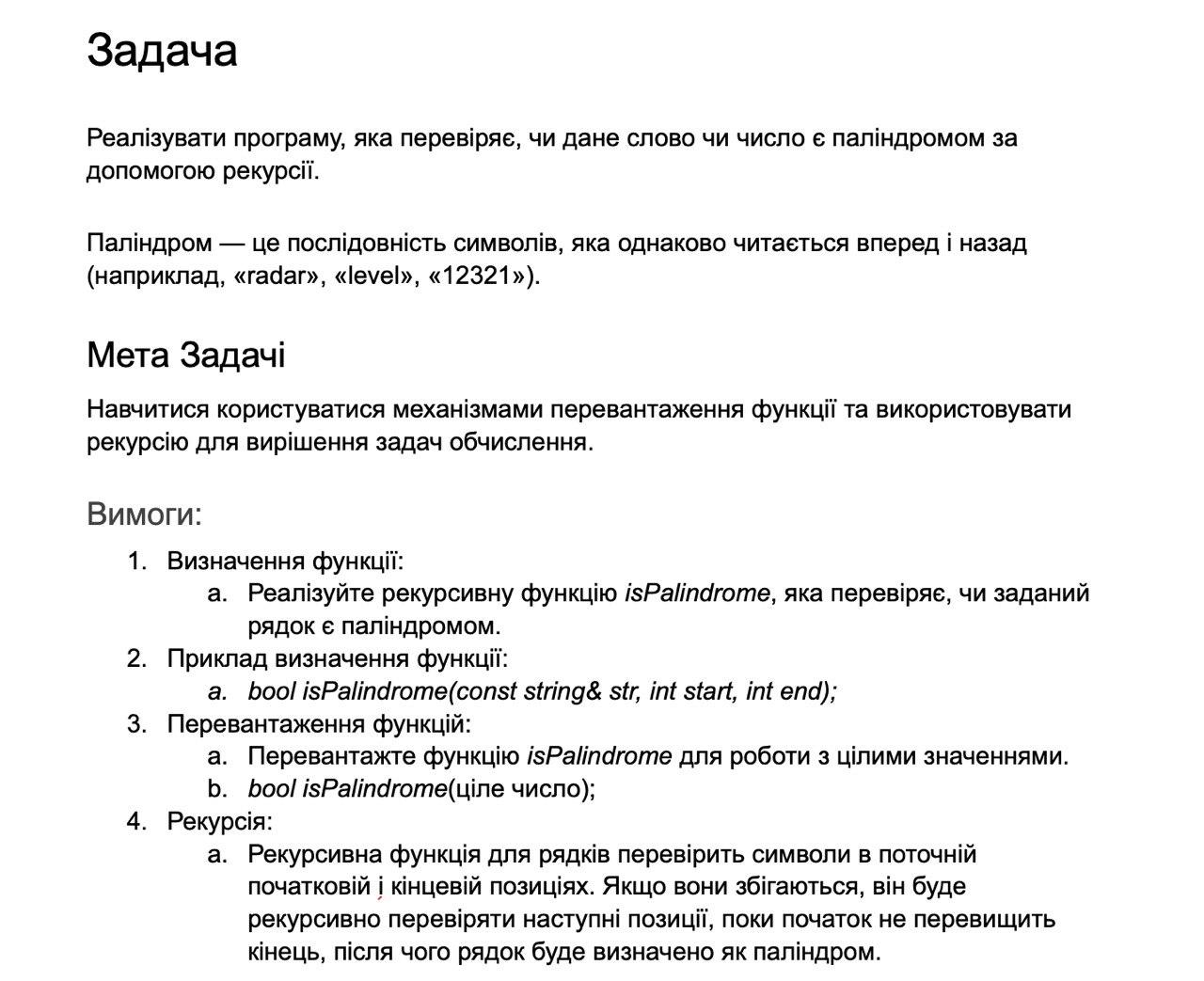
Завдання №4 Algotester Lab 3

* Варіант 3
* Деталі завдання



Завдання №5 Class Practice Work

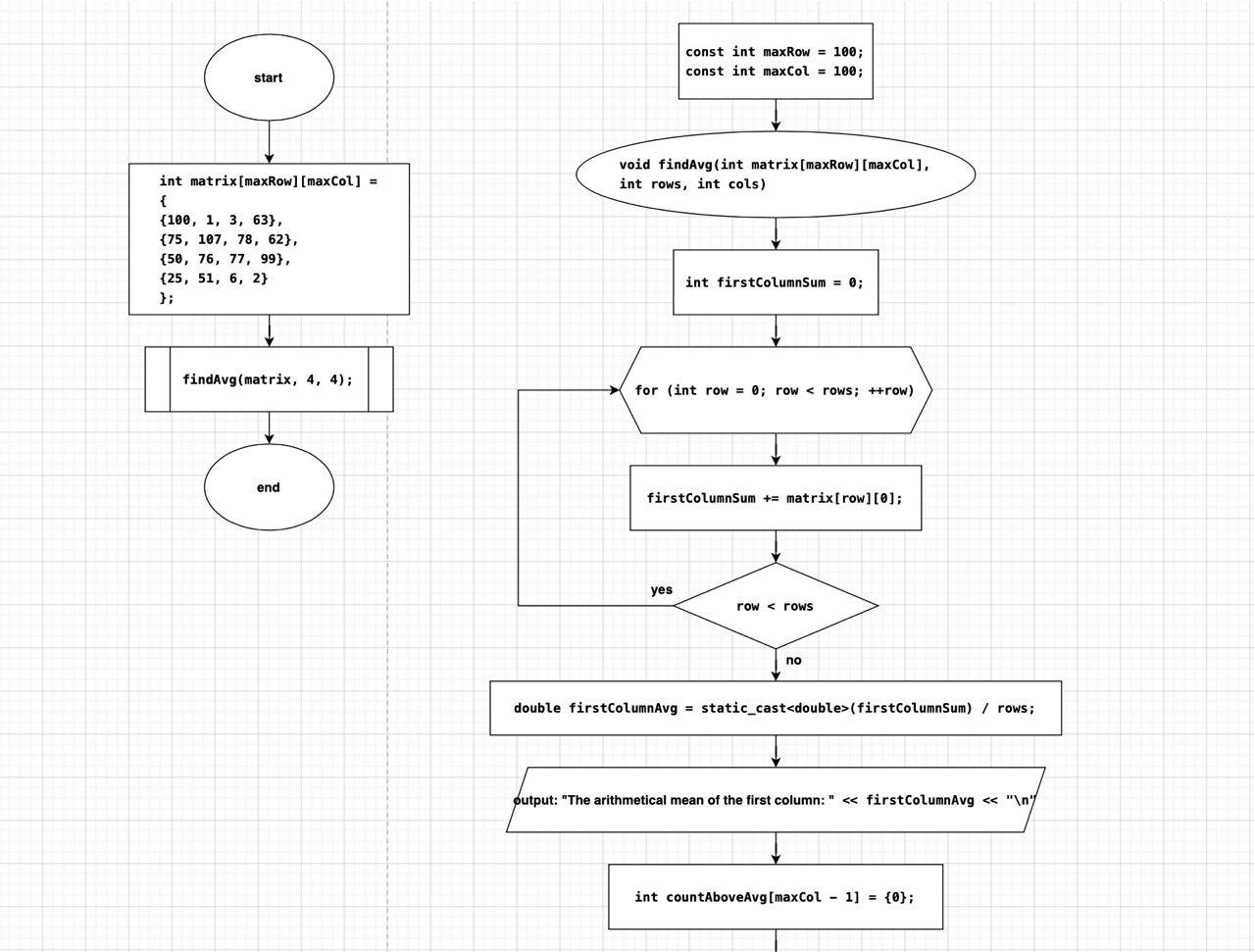
* Деталі завдання

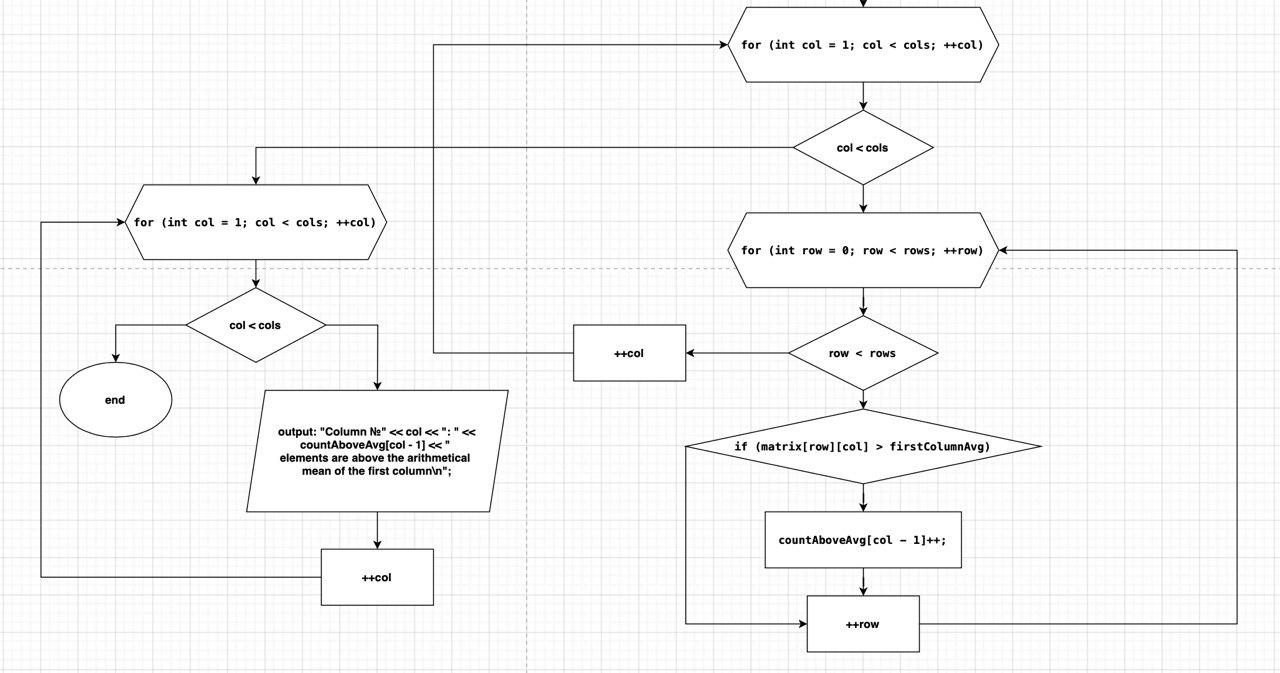


## **2. Дизайн та планована оцінка часу виконання завдань:**

Програма №2 VNS Lab 5

* Блок-схема





* Планований час на реалізацію 1 година

## **3. Код програм з посиланням на зовнішні ресурси:**

Завдання №1 VNS Lab 4

[vns\_lab\_4\_task\_vira\_mykhalchuk.cpp](https://github.com/artificial-intelligence-department/ai_programming_playground/blob/epic_4_practice_and_labs_vira_mykhalchuk/ai_14/vira_mykhalchuk/epic_4_practice_and_labs_vira_mykhalchuk/vns_lab_4_task_vira_mykhalchuk.cpp)

#include <iostream>

using namespace std;

const int n = 100;

int arr[n];

int top = -1;

int amountOfElements;

void destroy()

{

cout << "\n";

if(top >= 0)

{

cout << arr[top--] << " is now successfully deleted!\n";

}

else

{

cout << "You can't delete any element! Stack is empty.\n";

}

}

void output()

{

cout << "Current stack: ";

for(int i = 0; i <= top; i++)

{

cout << arr[i] << " " << "\n";

}

cout << "\n";

}

void add(int element)

{

cout << "\n";

if(top < amountOfElements-1)

{

arr[++top] = element;

cout << element << " is successfully added to the stack!\n";

}

else

{

cout << "The element " << element << " can't be added! Stack is full.\n";

}

}

int main()

{

int userInputSizeOfStack;

string userInputAction;

cout << "\nHi! Welcome to the stack! Enter the amount of elements of the stack (1;100): ";

cin >> userInputSizeOfStack;

while(userInputSizeOfStack > 100 or userInputSizeOfStack <=0)

{

cout << "The amount of elements is wrong! It must be greater than 0 and less than 100.\n";

cout << "Try again: ";

cin >> userInputSizeOfStack;

}

amountOfElements = userInputSizeOfStack;

do

{

cout << "You can:\n";

cout << "1. Add an element (type add)\n";

cout << "2. Delete an element (type delete)\n";

cout << "3. Exit the program (type exit)\n";

cout << "Enter your action here: ";

cin >> userInputAction;

switch(userInputAction[0])

{

case 'a':

int elementToAdd;

cout << "Enter the element you want to add: ";

cin >> elementToAdd;

add(elementToAdd);

output();

break;

case 'd':

destroy();

output();

break;

case 'e':

cout << "Goodbye! See you soon!" << "\n";

break;

default:

cout << "ERROR" << "\n";

break;

}

}

while(userInputAction != "exit");

return 0;

}

Завдання №2 VNS Lab 5

[vns\_lab\_5\_variant\_22\_vira\_mykhalchuk.cpp](https://github.com/artificial-intelligence-department/ai_programming_playground/blob/epic_4_practice_and_labs_vira_mykhalchuk/ai_14/vira_mykhalchuk/epic_4_practice_and_labs_vira_mykhalchuk/vns_lab_5_variant_22_vira_mykhalchuk.cpp)

#include <iostream>

using namespace std;

const int maxRow = 100;

const int maxCol = 100;

void findAvg(int matrix[maxRow][maxCol], int rows, int cols)

{

int firstColumnSum = 0;

for (int row = 0; row < rows; ++row)

{

firstColumnSum += matrix[row][0];

}

double firstColumnAvg = static\_cast<double>(firstColumnSum) / rows;

cout << "The arithmetical mean of the first column: " << firstColumnAvg << "\n";

int countAboveAvg[maxCol - 1] = {0};

for (int col = 1; col < cols; ++col)

{

for (int row = 0; row < rows; ++row)

{

if (matrix[row][col] > firstColumnAvg)

{

countAboveAvg[col - 1]++;

}

}

}

for (int col = 1; col < cols; ++col)

{

cout << "Column №" << col << ": " << countAboveAvg[col - 1] << " elements are above the arithmetical mean of the first column\n";

}

}

int main()

{

int matrix[maxRow][maxCol] =

{

{100, 1, 3, 63},

{75, 107, 78, 62},

{50, 76, 77, 99},

{25, 51, 6, 2}

};

findAvg(matrix, 4, 4);

return 0;

}

Завдання №3 Algotester Lab 2

[algotester\_lab\_2\_variant\_3\_vira\_mykhalchuk.cpp](https://github.com/artificial-intelligence-department/ai_programming_playground/blob/epic_4_practice_and_labs_vira_mykhalchuk/ai_14/vira_mykhalchuk/epic_4_practice_and_labs_vira_mykhalchuk/algotester_lab_2_variant_3_vira_mykhalchuk.cpp)

#include <iostream>

using namespace std;

int main() {

int N;

cin >> N;

int\* array = new int[N];

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

{

cin >> array[i];

}

int leftOne = 0;

int rightOne = 0;

int posLeft = 0;

int posRight = N - 1;

while (posLeft + 1 < posRight)

{

posLeft += leftOne;

posRight -= rightOne;

leftOne = array[posLeft];

rightOne = array[posRight];

}

cout << posLeft + 1 << " " << posRight + 1 << endl;

if (posLeft + 1 == posRight)

{

cout << "Stopped";

}

else if (posLeft == posRight)

{

cout << "Collision";

}

else

{

cout << "Miss";

}

return 0;

}

Завдання №4 Algotester Lab 3

[algotester\_lab\_3\_variant\_3\_vira\_mykhalchuk.cpp](https://github.com/artificial-intelligence-department/ai_programming_playground/blob/epic_4_practice_and_labs_vira_mykhalchuk/ai_14/vira_mykhalchuk/epic_4_practice_and_labs_vira_mykhalchuk/algotester_lab_3_variant_3_vira_mykhalchuk.cpp)

#include <iostream>

#include <string>

using namespace std;

int main()

{

string s;

cin >> s;

char current = s[0];

int count = 1;

for (int i = 1; i < s.size(); ++i)

{

if (s[i] == current)

{

count++;

}

else

{

cout << current;

if (count > 1)

{

cout << count;

}

current = s[i];

count = 1;

}

}

cout << current;

if (count > 1)

{

cout << count;

}

return 0;

}

Завдання №5 Class Practice Work

[class\_practice\_work\_vira\_mykhalchuk.cpp](https://github.com/artificial-intelligence-department/ai_programming_playground/blob/epic_4_practice_and_labs_vira_mykhalchuk/ai_14/vira_mykhalchuk/epic_4_practice_and_labs_vira_mykhalchuk/class_practice_work_vira_mykhalchuk.cpp)

#include <iostream>

#include <string>

#include <algorithm>

using namespace std;

bool isPalindrome(const string& str, int start, int end) {

if (start >= end) {

return true;

}

if (str[start] != str[end]) {

return false;

}

return isPalindrome(str, start + 1, end - 1);

}

bool isPalindrome(int num) {

string numStr = to\_string(num);

return isPalindrome(numStr, 0, numStr.length() - 1);

}

int main() {

string word;

cout << "Enter your word: ";

cin >> word;

if (isPalindrome(word, 0, word.length() - 1)) {

cout << word << " is a palidrome.\n";

} else {

cout << word << " isn't a palindrome.\n";

}

int number;

cout << "Enter your number: ";

cin >> number;

if (isPalindrome(number)) {

cout << number << " is a palidrome.\n";

} else {

cout << number << " isn't a palidrome.\n";

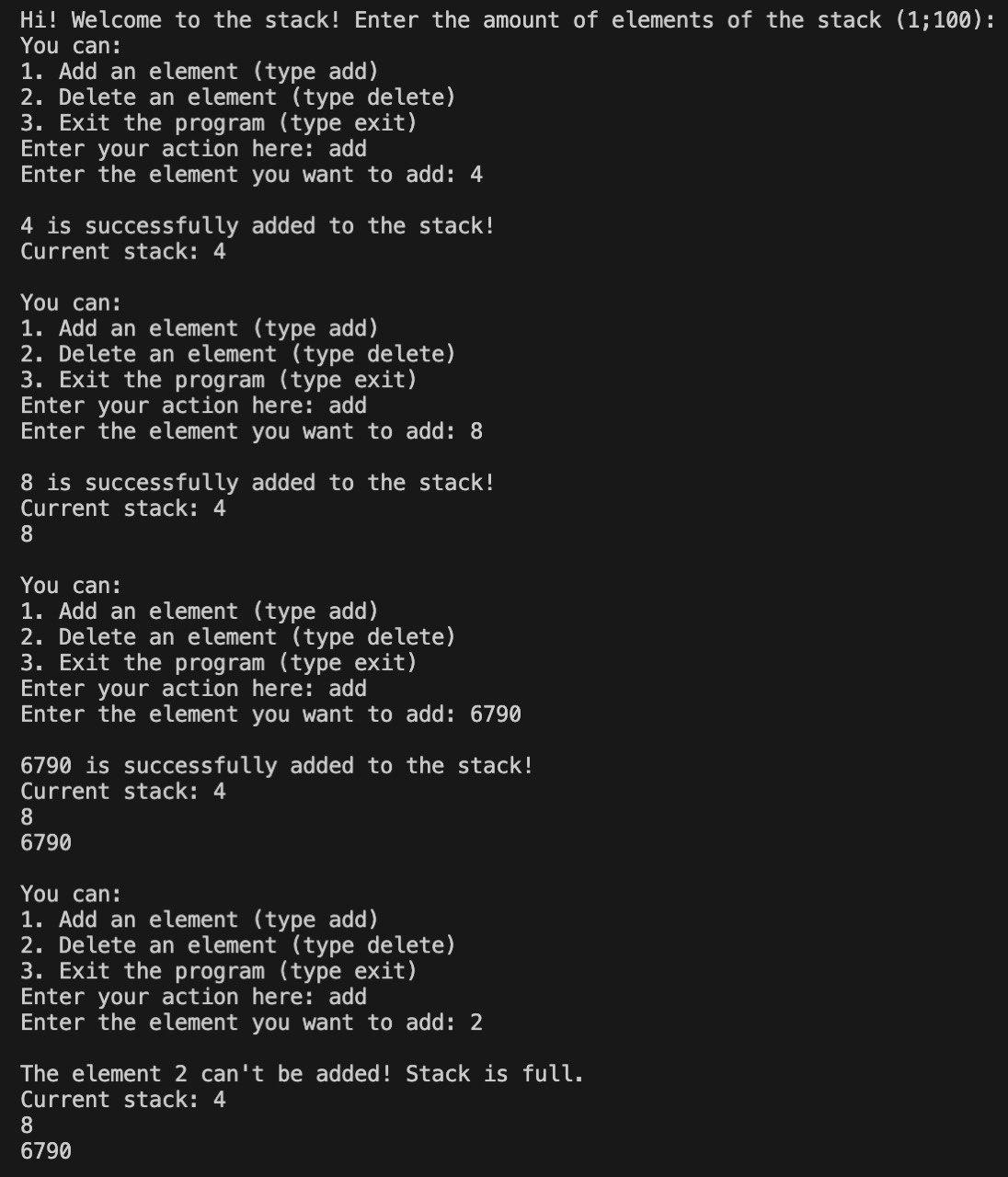
}

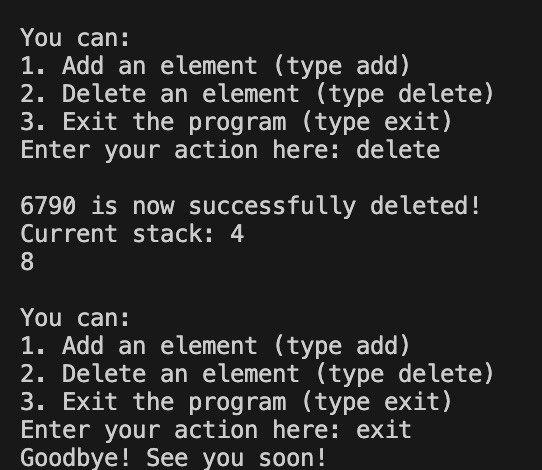
return 0;

}

## **5. Результати виконання завдань, тестування та фактично затрачений час:**

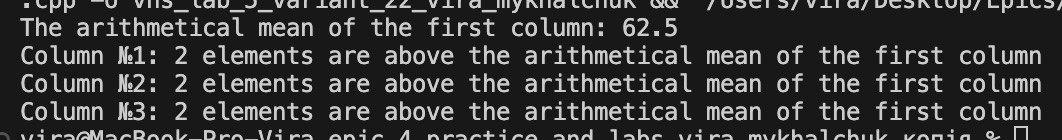
Завдання №1 VNS Lab 4





Час затрачений на виконання завдання 2,5 години

Завдання №2 VNS Lab 5



Час затрачений на виконання завдання 2 години

Завдання №3 Algotester Lab 2



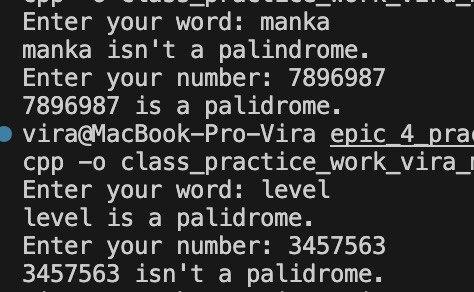
Час затрачений на виконання завдання 2,5 години

Завдання №4 Algotester Lab 3



Час затрачений на виконання завдання 2 години

Завдання №5 Class Practice Work



Час затрачений на виконання завдання 2,5 години

**Висновки:**

Під час вивчення тем, зокрема одновимірних та двовимірних масивів, а також алгоритмів обробки, було здобуто знання та навички у роботі з даними структурами даних.