

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



Звіт

про виконання лабораторних та практичних робіт блоку № 4

На тему: «Одновимірні масиви. Двовимірні Масиви. Вказівники та Посилання. Динамічні масиви. Структури даних. Вкладені структури. Алгоритми обробки та робота з масивами та структурами.»

з дисципліни: «Основи програмування»

до:

ВНС Лабораторної Роботи №4
ВНС Лабораторної Роботи №5
Алготестер Лабораторної Роботи №2
Алготестер Лабораторної Роботи №3
Практичних Робіт до блоку №4

Виконав:

Студент групи ШІ-11
Савков Олександр Андрійович

Львів 2024

Тема роботи:

Одновимірні масиви. Двовимірні Масиви. Вказівники та Посилання.
Динамічні масиви. Структури даних. Вкладені структури. Алгоритми
обробки та робота з масивами та структурами.»

Мета роботи:

Розібратись з класами пам'яті, зокрема зі статичною, динамічною пам'яттю, поняттям стеку, виділення та вивільнення пам'яті. Дослідження одновимірних і двовимірних масивів для зберігання і впорядкування даних, що забезпечує швидкий доступ і обробку великих обсягів інформації. Дослідження вказівників та посилань для розуміння адресації пам'яті та оптимізації використання ресурсів, що дозволяє ефективніше працювати з динамічними структурами даних. Дослідження динамічних масивів для створення програм із змінною кількістю елементів, що підвищує гнучкість і адаптивність коду. Дослідження структур даних та вкладених структур для організації складних об'єктів, що забезпечує кращу структуру і читабельність програмного коду. Дослідження алгоритмів обробки масивів і структур для реалізації ефективної обробки даних, що сприяє написанню оптимізованих і масштабованих програм.

Теоретичні відомості:

1. Індивідуальний план опрацювання теорії:

Тема № 1: Введення в класи пам'яті у C++

- **Джерела інформації:**
 - Статті:
 - [Класи пам'яті у C++](#)
 - Відео:
 - [Основи статичної та динамічної пам'яті](#)
- **Що опрацьовано:**
 - Статична пам'ять.
 - Динамічна пам'ять.
 - Поняття стеку.
 - Виділення та вивільнення пам'яті.
- **Запланований час на вивчення і пошук:** Витрачений час 1 година.

Тема № 2: Вступ до масивів і вказівників

- **Джерела інформації:**
 - Статті:
 - Основи масивів та вказівників у C++
 - Відео:
 - [Масиви та вказівники: Основи](#)

- **Що опрацьовано:**
 - Основи масивів: визначення, важливість, приклади використання.
 - Різниця між статичними та динамічними масивами.
 - Основи вказівників: що це таке, як вони працюють.
 - Взаємоз'язок між масивами та вказівниками.
 - Вступ до посилань: основні концепції та відмінності від вказівників.
- **Запланований час на вивчення і пошук:** Витрачений час 1 година.

Тема № 3: Одновимірні масиви

- **Джерела інформації:**
 - Статті:
 - Створення та ініціалізація одновимірних масивів
 - Відео:
 - [Одновимірні масиви та їх обхід](#)
- **Що опрацьовано:**
 - Створення та ініціалізація одновимірних масивів.
 - Основні операції: індексація, присвоєння, читання.
 - Цикли та обхід масивів.
 - Використання функцій для роботи з масивами.
 - Приклади алгоритмів сортування та пошуку.
- **Запланований час на вивчення і пошук:** Витрачений час 2 години.

Тема № 4: Вказівники та посилання

- **Джерела інформації:**
 - Статті:
 - Вказівники та посилання у C++
 - Відео:
 - [Основи вказівників та посилань](#)
- **Що опрацьовано:**
 - Використання вказівників для доступу до елементів масиву.
 - Арифметика вказівників.
 - Різниця між вказівниками та посиланнями в контексті функцій.
 - Динамічне виділення пам'яті з використанням вказівників.
 - Використання вказівників для створення складних структур даних.
- **Запланований час на вивчення і пошук:** Витрачений час 2 години.

Тема № 5: Двовимірні масиви

- **Джерела інформації:**
 - Статті:
 - Двовимірні масиви в C++
 - Відео:
 - [Двовимірні масиви та їх обхід](#)
- **Що опрацьовано:**

- Оголошення та ініціалізація двовимірних масивів.
- Вкладені цикли для обходу двовимірних масивів.
- Практичні приклади використання двовимірних масивів.
- Передача двовимірних масивів у функції.
- Застосування двовимірних масивів для розв'язання задач.
- **Запланований час на вивчення і пошук:** Витрачений час 3 години.

Тема № 6: Динамічні масиви

- **Джерела інформації:**
 - Статті:
 - Динамічні масиви в C++
 - Відео:
 - [Динамічне виділення пам'яті для масивів](#)
- **Що опрацьовано:**
 - Основи динамічного виділення пам'яті.
 - Створення та управління динамічними масивами.
 - Використання операторів `new` та `delete` для управління пам'яттю.
 - Реалізація змінної розмірності масивів.
 - Передача динамічних масивів у функції.
- **Запланований час на вивчення і пошук:** Витрачений час 2 години.

Тема № 7: Структури даних

- **Джерела інформації:**
 - Статті:
 - Структури даних у C++
 - Відео:
 - [Основи структур даних у C++](#)
- **Що опрацьовано:**
 - Оголошення та використання структур.
 - Використання масивів та вказівників у структурах.
 - Функції для обробки даних у структурах.
 - Використання структур для представлення складних даних.
 - Вкладені структури та їх використання.
 - Об'єднання (Union) та переліки (enumerations).
- **Запланований час на вивчення і пошук:** Витрачений час 3 години.

Тема № 8: Вкладені структури

- **Джерела інформації:**
 - Статті:
 - Вкладені структури в C++
 - Відео:
 - [Робота з вкладеними структурами](#)
- **Що опрацьовано:**

- Поняття вкладених структур та їх оголошення.
- Взаємодія з вкладеними структурами.
- Використання вкладених структур для моделювання складних даних.
- Передача вкладених структур у функції.
- Приклади реального використання вкладених структур.
- **Запланований час на вивчення і пошук:** Витрачений час 2 години.

Тема № 9: Використання структур

- **Джерела інформації:**
 - Статті:
 - Перевантаження операторів у структурах
 - Відео:
 - [Оператори виводу/вводу та арифметичні операції зі структурами](#)
- **Що опрацьовано:**
 - Перевантаження операторів у структурі.
 - Вивід/ввід структури (`operator<<`).
 - Арифметичні операції з структурами (`operator+`, `operator-`).
 - Практичні задачі на виведення структур та операції з ними.
- **Запланований час на вивчення і пошук:** Витрачений час 3 години.

Тема № 10: Алгоритми обробки та робота з масивами та структурами

- **Джерела інформації:**
 - Статті:
 - Алгоритми роботи з масивами та структурами
 - Відео:
 - [Пошук та сортування в масивах](#)
- **Що опрацьовано:**
 - Алгоритми пошуку та сортування в масивах.
 - Обробка та маніпуляції з даними у структурах.
 - Використання циклів та умовних операторів для роботи з масивами та структурами.
 - Інтеграція масивів та структур у алгоритми.
 - Розв'язання практичних задач з використанням масивів та структур.
- **Запланований час на вивчення і пошук:** Витрачений час 4 години.

Виконання роботи:

1. Опрацювання завдання та вимог до програм:

VNS Lab 4 - Task 1 Варіант 3

3.

- 1) Сформувати одновимірний масив цілих чисел, використовуючи генератор випадкових чисел.
- 2) Роздрукувати отриманий масив.
- 3) Знищити всі елементи, які дорівнюють 0.
- 4) Додати після першого парного елемента масиву елемент зі значенням $M[I-1]+2$.
- 5) Роздрукувати отриманий масив.

VNS Lab 5 - Task 1 Варіант 3

3. Написати процедуру для підсумовування матриць. З її допомогою скласти вихідну матрицю й транспоновану (тобто отриману поворотом вихідної на 90°).

Algotester Lab 2 Варіант 1

Lab 2v1

Обмеження: 1 сек., 256 MiB

У вас є дорога, яка виглядає як N чисел.

Після того як ви по ній пройдете - вашу втому можна визначити як різницю максимального та мінімального елементу.

Ви хочете мінімізувати втому, але все що ви можете зробити - викинути одне число з дороги, тобто забрати його з масиву.

В результаті цієї дії, яку мінімальну втому ви можете отримати в кінці дороги?

Вхідні дані

У першому рядку ціле число N - кількість чисел

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

Вихідні дані

Єдине ціле число m - мінімальна втома, яку можна отримати

Обмеження

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

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

Algotester Lab 3 Варіант 2

Lab 3v2

Обмеження: 1 сек., 256 MiB

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).

Обмеження

$$0 \leq N \leq 100$$

$$0 \leq a_i \leq 100$$

$$0 \leq M \leq 100$$

$$0 \leq b_i \leq 100$$

Class Practice work

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:

- Визначення функції:
 - Реалізуйте рекурсивну функцію *isPalindrome*, яка перевіряє, чи заданий рядок є паліндромом.
- Приклад визначення функції:
 - `bool isPalindrome(const string& str, int start, int end);`
- Перевантаження функцій:
 - Перевантажте функцію *isPalindrome* для роботи з цілими значеннями.
 - `bool isPalindrome(ціле число);`
- 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. Використати математичний підхід щоб перевірити чи число є паліндромом.

Self practice work algotester

Депутатські гроші

Обмеження: 2 сек., 256 MiB

Часто-густо громадяни намагаються з'ясувати, наскільки багатими є депутати. Дехто вірить, що матеріальні статки окремих депутатів є необмеженими.

Тож уявіть собі депутата, у якого є необмежена кількість купюр усіх номіналів (1, 2, 5, 10, 20, 50, 100, 200 та 500 гривень). Він хоче придбати подарунок для своєї тещі, що коштує n гривень.

Незважаючи на те, що наш герой-олігарх є нескінченно багатим, він також є нескінченно скупим та педантичним. Саме тому він хоче оплатити покупку готівкою без решти. Яка мінімальна кількість купюр йому для цього знадобиться?

Вхідні дані

У єдиному рядку задано одне натуральне число n — вартість подарунку.

Вихідні дані

У єдиному рядку виведіть одне ціле число — мінімальну кількість купюр, що необхідна для покупки подарунка.

Обмеження

$1 \leq n \leq 10^9$.

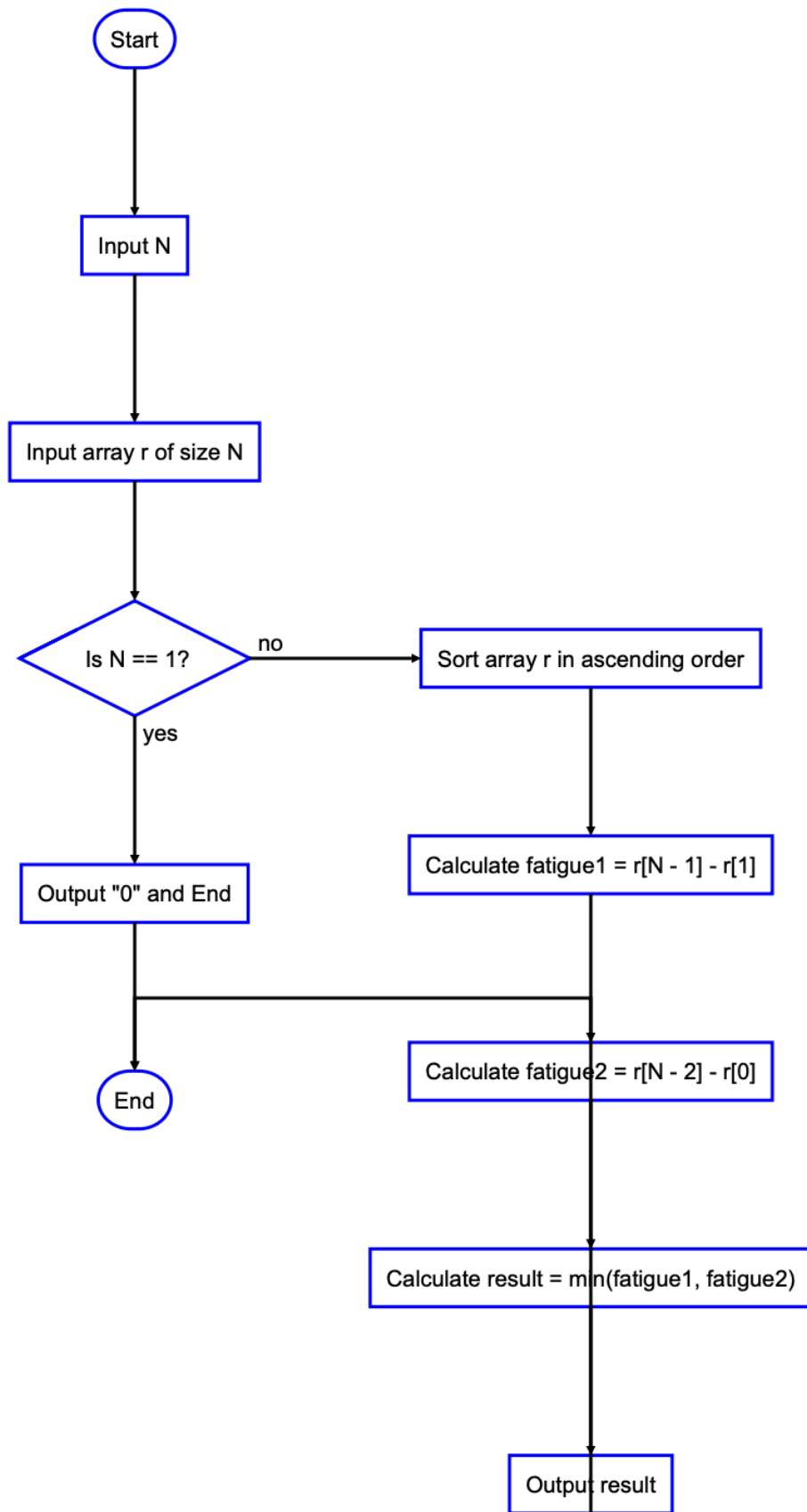
Приклади

2. Дизайн та планована оцінка часу виконання завдань:

Self practice work algotester

Дизайн алгоритму завдання знаходиться в папці “Flowcharts” під назвою algotester_lab_2_variant_1_oleksandr_savkov.pdf

Попередньо розраховував, що це завдання займе 10 хв



3. Код програм з посиланням на зовнішні ресурси

VNS Lab 4 - Task 1 Варіант 3

```
#include <iostream>
#include <cstdlib>
#include <ctime>

using namespace std;

int main(){
    srand(static_cast<unsigned>(time(0)));
    int kilkistEl = 0;
    cout << "Введіть кількість елементів в масиві" << endl;
    cin >> kilkistEl;
    if (kilkistEl >= 100) {
        cout << "Помилка: кількість елементів не може перевищувати 100, спробуйте ще раз" << endl;
        cin >> kilkistEl;
    }
    int mainArr[100];
    cout << "Масив: " << endl;
    for (int i = 0; i < kilkistEl; i++){
        mainArr[i] = rand() % 100 + 1;
        cout << mainArr[i] << endl;
    }
    cout << endl;

    for(int i = 0; i < kilkistEl; ) {
        if (mainArr[i] == 0) {
            for (int j = i; j < kilkistEl - 1; j++) {
                mainArr[j] = mainArr[j + 1];
            }
            kilkistEl--;
        } else {
            i++;
        }
    }

    bool key = true;
    for(int i = 0; i < kilkistEl; i++){
        if(mainArr[i] % 2 == 0 && key) {
            for(int k = kilkistEl; k > i + 1; k--) {
                mainArr[k] = mainArr[k - 1];
            }
            mainArr[i + 1] = mainArr[i] + 2;
            kilkistEl++;
            key = false;
        }
    }

    cout << "Відредагований масив: " << endl;
    for (int i = 0; i < kilkistEl; i++) {
        cout << mainArr[i] << endl;
    }
    cout << endl;

    return 0;
}
```

Код знаходиться в папці “Code” під назвою

vns_lab_4_task_1_variant_3_oleksandr_savkov.cpp

VNS Lab 5 - Task 1 Варіант 3

```
11 > savkov_oleksandr > epic_4 > Code > vns_lab_5_task_1_variant_3_oleksandr_savkov.cpp
1  #include <iostream>
2  using namespace std;
3
4  int main() {
5      int width, height;
6
7      cout << "Введіть кількість рядків матриці: ";
8      cin >> height;
9
10     cout << "Введіть кількість стовпців матриці: ";
11     cin >> width;
12
13     int matrix[100][100];
14     cout << "Введіть елементи матриці: " << endl;
15     for (int i = 0; i < height; i++) {
16         for (int k = 0; k < width; k++) {
17             cout << "Елемент [" << i + 1 << "]" << k + 1 << "]: ";
18             cin >> matrix[i][k];
19         }
20     }
21
22     cout << "Вихідна матриця: " << endl;
23     for (int i = 0; i < height; i++) {
24         for (int j = 0; j < width; j++) {
25             cout << matrix[i][j] << " ";
26         }
27         cout << endl;
28     }
29
30     cout << "Транспонована матриця: " << endl;
31     for (int i = 0; i < width; i++) {
32         for (int j = 0; j < height; j++) {
33             cout << matrix[j][i] << " ";
34         }
35         cout << endl;
36     }
37
38     return 0;
39 }
```

Код знаходиться в папці “Code” під назвою

vns_lab_5_task_1_variant_3_oleksandr_savkov.cpp

Algotester Lab 2 Варіант 1

```
#include <iostream>
#include <vector>
#include <algorithm>

using namespace std;

int minimize_fatigue(int N, vector<int>& r) {
    if (N == 1) return 0;
    sort(r.begin(), r.end());
    int fatigue1 = r[N - 1] - r[1];
    int fatigue2 = r[N - 2] - r[0];
    return min(fatigue1, fatigue2);
}

int main() {
    int N;
    cin >> N;
    vector<int> r(N);
    for (int i = 0; i < N; i++) cin >> r[i];
    cout << minimize_fatigue(N, r) << endl;
    return 0;
}
```

Код знаходиться в папці “Code” під назвою
algotester_lab_2_variant_1_oleksandr_savkov.cpp

Algotester Lab 3 Варіант 2

```

savkov_oleksandr / epi_1 / Code / algotester_lab_3_variant_2_oleksandr_savkov.cpp / main()
#include <iostream>
#include <vector>
#include <algorithm>

using namespace std;

int main() {
    int n, m;
    cin >> n;
    vector<int> arrFirst(n);
    for (int i = 0; i < n; i++) cin >> arrFirst[i];

    cin >> m;
    vector<int> arrSecond(m);
    for (int i = 0; i < m; i++) cin >> arrSecond[i];

    vector<int> commonElements;
    for (int i = 0; i < n; i++) {
        for (int j = 0; j < m; j++) {
            if (arrFirst[i] == arrSecond[j]) {
                commonElements.push_back(arrFirst[i]);
                break;
            }
        }
    }

    sort(commonElements.begin(), commonElements.end());
    commonElements.erase(unique(commonElements.begin(), commonElements.end()), commonElements.end());
    int commonCount = commonElements.size();

    vector<int> allElements = arrFirst;
    allElements.insert(allElements.end(), arrSecond.begin(), arrSecond.end());

    sort(allElements.begin(), allElements.end());
    allElements.erase(unique(allElements.begin(), allElements.end()), allElements.end());
    int uniqueCount = allElements.size();

    cout << commonCount << endl;
    cout << uniqueCount << endl;

    return 0;
}

```

Код знаходиться в папці “Code” під назвою
algotester_lab_3_variant_2_oleksandr_savkov.cpp

Class Practice work

```

savkov_oleksandr / epic_4 / code / C++ / practice_work_task_1_oleksandr_savkov.cpp / num
#include <iostream>
#include <string>
#include <algorithm>

using namespace std;

bool numCount(const string& word, int start, int end) {
    if (start >= end) {
        return true;
    }
    if (word[start] != word[end]) {
        return false;
    }
    return numCount(word, start + 1, end - 1);
}

bool isPolindrom(const string& word) {
    return numCount(word, 0, word.size() - 1);
}

bool isPolindrom(int number) {
    string numberStr = to_string(number);
    return numCount(numberStr, 0, numberStr.size() - 1);
}

int main(){
    string word;
    cout << "Введіть слово для перевірки: ";
    cin >> word;
    if (isPolindrom(word))
    {
        cout << "Слово є паліндромом." << endl;
    }else{
        cout << "Слово не є паліндромом." << endl;
    }

    int number;
    cout << "Введіть число: ";
    cin >> number;

    if (isPolindrom(number))
    {
        cout << "Число є паліндромом." << endl;
    }else{
        cout << "Число не є паліндромом." << endl;
    }
    return 0;
}

```

Код знаходиться в папці “Code” під назвою
practice_work_task_1_oleksandr_savkov.cpp

Self practice work algotester

```
#include <iostream>
#include <vector>

using namespace std;

int main() {
    vector<int> kupiurs = {500, 200, 100, 50, 20, 10, 5, 2, 1};
    long long a;
    long long result = 0;
    cin >> a;
    for (int i = 0; i < kupiurs.size(); i++) {
        result += a / kupiurs[i];
        a %= kupiurs[i];
    }
    cout << result << endl;
    return 0;
}
```

Код знаходиться в папці “Code” під назвою
self_practice_work_algotester_task_1_oleksandr_savkov.cpp

Результати виконання завдань, тестування та фактично затрачений час:

VNS Lab 4 - Task 1 Варіант 3

```
mikamika — 80x24
r_savkov'
Working directory: '/Users/mikamika/Documents/osnovy_program/epics/ai_programmin
g_playground_2024/ai_11/savkov_oleksandr/epic_4/Code'
1 arguments:
argv[0] = '/Users/mikamika/Documents/osnovy_program/epics/ai_programming_playgro
und_2024/ai_11/savkov_oleksandr/Epic_4/Code/vns_lab_4_task_1_variant_3_oleksandr
_savkov'
Введіть кількість елементів в масиві
3
Масив:
37
97
16

Відредагований масив:
37
97
16
18

Process exited with status 0

Saving session...
...copying shared history...
```

Зайняло часу: 30 хв

VNS Lab 5 - Task 1 Варіант 3

```
mikamika — 80x38

Launching: '/Users/mikamika/Documents/osnovy_program/epics/ai_progr
ound_2024/ai_11/savkov_oleksandr/Epic_4/Code/vns_lab_5_task_1_vari
r_savkov'
Working directory: '/Users/mikamika/Documents/osnovy_program/epics
g_playground_2024/ai_11/savkov_oleksandr/epic_4/Code'
1 arguments:
argv[0] = '/Users/mikamika/Documents/osnovy_program/epics/ai_progr
ound_2024/ai_11/savkov_oleksandr/Epic_4/Code/vns_lab_5_task_1_varia
_savkov'
Введіть кількість рядків матриці: 3
Введіть кількість стовпців матриці: 3
Введіть елементи матриці:
Елемент [1][1]: 2
Елемент [1][2]: 3
Елемент [1][3]: 6
Елемент [2][1]: 5
Елемент [2][2]: 3
Елемент [2][3]: 3
Елемент [3][1]: 4
Елемент [3][2]: 2
Елемент [3][3]: 9
Вихідна матриця:
2 3 6
5 3 3
4 2 9
Транспонована матриця:
2 5 4
3 3 2
6 3 9
Process exited with status 0

Saving session...
...copying shared history...
...saving history...truncating history files...
...completed.

[Process completed]
```

Зайняло часу: 30 хв

Algotester Lab 2 Варіант 1

```
Launching: '/Users/mikamika/Documents/osnound_2024/ai_11/savkov_oleksandr/Epic_4/r_savkov'
Working directory: '/Users/mikamika/Documents/osnound_2024/ai_11/savkov_oleksandr'
1 arguments:
argv[0] = '/Users/mikamika/Documents/osnound_2024/ai_11/savkov_oleksandr/Epic_4/C_r_savkov'
3
4
5
6
1
Process exited with status 0

Saving session...
...copying shared history...
...saving history...truncating history file...completed.

[Process completed]
```

Зайняло часу	Тема	Статус
4 години тому	C++ 23	Зараховано

Зайняло часу: 1 година

Algotester Lab 3 Варіант 2

```
ound_2024/ai_11/savkov
r_savkov'
Working directory: '/U
g_playground_2024/ai_1
1 arguments:
argv[0] = '/Users/mika
und_2024/ai_11/savkov_
_savkov'
3
3
4
5
2
3
4
2
3
Process exited with st

Saving session...
...copying shared hist
...saving history...tr
...completed.
```

Створено	Компілятор	Результат
3 години тому	C++ 23	Зараховано

Зайняло часу: 1 година

Class Practice work

```
mikamika — 80x24
Launching: '/Users/mikamika/Documents/osnovy_program/epics/ai_programming_playground_2024/ai_11/savkov_oleksandr/epic_4/Code/practice_work_task_1_oleksandr_savkov'
Working directory: '/Users/mikamika/Documents/osnovy_program/epics/ai_programming_playground_2024/ai_11/savkov_oleksandr/epic_4/Code'
1 arguments:
argv[0] = '/Users/mikamika/Documents/osnovy_program/epics/ai_programming_playground_2024/ai_11/savkov_oleksandr/epic_4/Code/practice_work_task_1_oleksandr_savkov'
Введіть слово для перевірки: rebber
Слово є паліндромом.
Введіть число: 123321
Число є паліндромом.
Process exited with status 0

Saving session...
...copying shared history...
...saving history...truncating history files...
...completed.

[Process completed]
```

Зайняло часу: 30 хв

Self practice work algotester

```
argv[0] = '/Users/mikamika/Documents/osnovy_program/epics/ai_programming_playground_2024/ai_11/savkov_oleksandr/epic_4/Code/practice_work_task_1_oleksandr_savkov'
86
5
Process exited with status 0

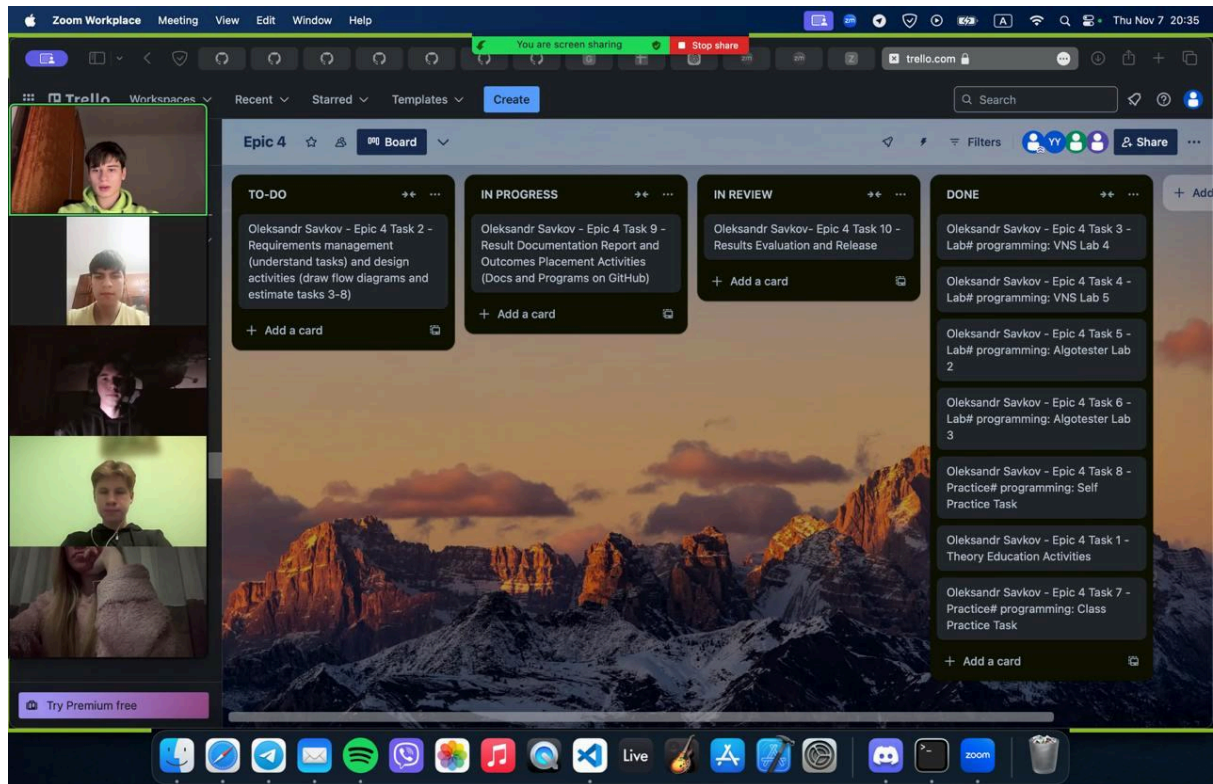
Saving session...
...copying shared history...
...saving history...truncating history files...
...completed.

[Process completed]
```

Створено	Компілятор	Результат	Час (сек.)
43 хвилини тому	C++ 23	Зараховано	0.003

Зайняло часу: 10 хв

4. Кооперація з командою



Висновок: Під час виконання цієї роботи я вивчив багато нового, зокрема дізнався, що таке рекурсія і як її застосовувати, навчився використовувати одновимірні і двовимірні масиви. Також дізнався, що таке вказівники і посилання та яка між ними різниця і що краще застосовувати, щоб ефективніше користуватись і керувати пам'яттю. Ще я навчився використовувати динамічні масиви, зрозумів, що таке динамічна пам'ять.