

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



Звіт

про виконання лабораторних та практичних робіт блоку № 4
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-12
Іваник Тарас Юрійович

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:

У цій роботі розглядаються основні принципи роботи з масивами та структурами даних, зокрема одновимірні й двовимірні масиви для організації і зберігання великих обсягів даних. Особливу увагу приділено вказівникам і посиланням як засобам управління пам'яттю та ефективного доступу до даних. Розглянуто динамічні масиви, які забезпечують гнучке управління розміром даних під час виконання програми. Досліджено основи структур даних і вкладених структур для створення складних, логічно організованих об'єктів. Описано алгоритми обробки масивів і структур, що дозволяють ефективно виконувати операції пошуку, сортування і модифікації даних, покращуючи оптимізацію коду.

Джерела:

- 1) University lectures
- 2) Acode
- 3) W3 Schools
- 4) [C++ tutorial on YouTube \(BroCode\)](#)

Lab# programming: VNS Lab 4 (50 хв)

8.

- 1) Сформувати одновимірний масив цілих чисел, використовуючи генератор випадкових чисел.
- 2) Роздрукувати отриманий масив.
- 3) Знищити всі елементи із заданим значенням.
- 4) Додати перед кожним парним елементом масиву елемент зі значенням 0.
- 5) Роздрукувати отриманий масив.

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

using namespace std;

int main() {
    const int n = 10;
    int arr[n];
    int delete_value;

    srand(time(0));

    cout << "Start array: ";
    for (int i = 0; i < n; ++i){
        arr[i] = rand() % 10;
        cout << arr[i] << " ";
    }
    cout << "\n";

    cout << "Enter a value that we should delete: ";
    cin >> delete_value;

    int newSize = 0;
    for (int i = 0; i < n; ++i){
        if(arr[i] != delete_value){
            arr[newSize++] = arr[i];
        }
    }

    int newArr[2 * newSize]; // maximum size of new array with 0 before pair integers
    int NEWSIZE = 0;

    for (int i = 0; i < newSize; ++i){
        if(arr[i] % 2 == 0 && arr[i] != 0){
            newArr[NEWSIZE++] = 0;
        }
        newArr[NEWSIZE++] = arr[i];
    }
    cout << "Array after deleting numbers and adding zeros: ";
    for (int i = 0; i < NEWSIZE; ++i){
        cout << newArr[i] << " ";
    }
    cout << "\n";
    return 0;
}
```

```
Start array: 1 4 1 9 7 2 6 7 6 1
Enter a value that we should delete: 6
Array after deleting numbers and adding zeros: 1 0 4 1 9 7 0 2 7 1
PS C:\Users\User>
```

Lab# programming: VNS Lab 5 (140 хв)

8. Написати функцію транспонування квадратної матриці (тобто повороту початкової матриці на 90°). З її допомогою визначити чи є задана матриця симетричною. (Матриця називається симетричною, якщо транспонована матриця дорівнює початковій).

```
#include <iostream>
using namespace std;

void transpose(int n, int matrix[][100]){
    for (int i = 0; i < n; ++i){
        for(int j = 0; j < n; ++j){
            swap(matrix[i][j], matrix[j][i]);
        }
    }
}

bool isSymetric(int n, int matrix[][100]){
    transpose(n, matrix);

    for (int i = 0; i < n; ++i){
        for (int j = 0; j < n; ++j){
            if(matrix[i][j] != matrix[j][i]){
                return false;
            }
        }
    }
    return true;
}

int main(){
    int n;
    cout << "Enter size of matrix n x n" << endl;
    cin >> n;

    int matrix[100][100];

    cout << "Enter elements of this matrix: " << endl;
    for (int i = 0; i < n; ++i){
        for (int j = 0; j < n; ++j){
            cin >> matrix[i][j];
        }
    }
    if (isSymetric(n, matrix)){
        cout << "Your matrix is symetric!" << endl;
    } else{
        cout << "Your matrix is not symetric!" << endl;
    }
    return 0;
}
```

```
Enter size of matrix n x n
4
Enter elements of this matrix:
1 0 4 5
0 1 5 4
4 5 1 0
5 4 0 1
Your matrix is symetric!
```

Lab# programming: Algotester Lab 2 (120 xB)

```
#include <iostream>
#include <climits>

using namespace std;

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

    if(N <= 1){
        cout << 0;
        return 0;
    }
    int r[N];

    for (int i = 0; i < N; ++i) {
        cin >> r[i];
    }

    int max1 = INT_MIN, max2 = INT_MIN;
    int min1 = INT_MAX, min2 = INT_MAX;

    for (int i = 0; i < N; ++i){
        if(r[i] < min1){
            min2 = min1;
            min1 = r[i];
        } else if (r[i] < min2){
            min2 = r[i];
        }
        if(r[i] > max1){
            max2 = max1;
            max1 = r[i];
        } else if(r[i] > max2){
            max2 = r[i];
        }
    }

    if(max1 - min2 <= max2 - min1){
        M = max1 - min2;
    } else {
        M = max2 - min1;
    }
    cout << M;

    return 0;
}
```

Lab# programming: Algotester Lab 3 (80 xB)

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

using namespace std;

int main() {
    int N, a, b, c;
    cin >> N;

    vector<int> r(N);

    for (int i = 0; i < N; ++i) {
        cin >> r[i];
    }
    cin >> a >> b >> c;

    r.erase(remove(r.begin(), r.end(), a), r.end());
    r.erase(remove(r.begin(), r.end(), b), r.end()); //Видаляє з вектора a,b,c
    r.erase(remove(r.begin(), r.end(), c), r.end());

    if (r.size() < 2) {
        cout << 0 << endl;
        return 0;
    }

    vector<int> sumArray;
    for (int i = 0; i < r.size() - 1; ++i) {
        sumArray.push_back(r[i] + r[i + 1]);
    }
    cout << sumArray.size() << endl;
    for (int num : sumArray) {
        cout << num << " ";
    }
    cout << endl;

    return 0;
}
```

3 hours ago

Lab 3v3 - Lab 3v3

C++ 23

Accepted

0.003

1.180

1859977

Practice# programming: Class Practice Task (80 хв)

```
#include <iostream>
#include <string>

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); // наступна пара значень (рухаємось від кінців до середини)
}

bool isPalindrome(int number) {

    if (number < 0) return false; // тому що (-) може стояти тільки з лівого боку

    int reversed = 0;
    int original = number;

    while (number != 0) {
        int digit = number % 10;
        reversed = reversed * 10 + digit;
        number /= 10;
    }
    return original == reversed;
}

int main() {
    string str;
    int num;

    cout << "Input a string ";
    getline(cin, str);
    cout << "\n";
    cout << "Input a number ";
    cin >> num;
    cout << "\n";

    cout << "Is your string palindrome?: " << (isPalindrome(str, 0, str.length()-1) ? "Yes" : "No") << endl;
    cout << "Is your integer palindrome?: " << (isPalindrome(num) ? "Yes" : "No") << endl;

    return 0;
}
```

Input a string pylyp

Input a number 123321

Is your string palindrome?: Yes

Is your integer palindrome?: Yes

PS C:\Users\User> ^C

PS C:\Users\User>

PS C:\Users\User> & 'c:\Users\Us
bmh4enev.wc4' '--stderr=Microsoft

Input a string trasa

Input a number 12345464

Is your string palindrome?: No

Is your integer palindrome?: No

Practice# programming: Self Practice Task (30 хв)

```
/**Депутатські гроші (Algotester)***/
#include <iostream>
using namespace std;

int minBanknotes(int n) {
    int banknotes[] = {500, 200, 100, 50, 20, 10, 5, 2, 1};
    int count = 0;

    for (int banknote : banknotes) {
        count += n / banknote;
        n %= banknote;
    }

    return count;
}

int main() {
    int n;
    cin >> n;
    cout << minBanknotes(n) << endl;
    return 0;
}
```

1999
11

245
4

546
5

43 minutes ago

0021 - Депутатські гроші

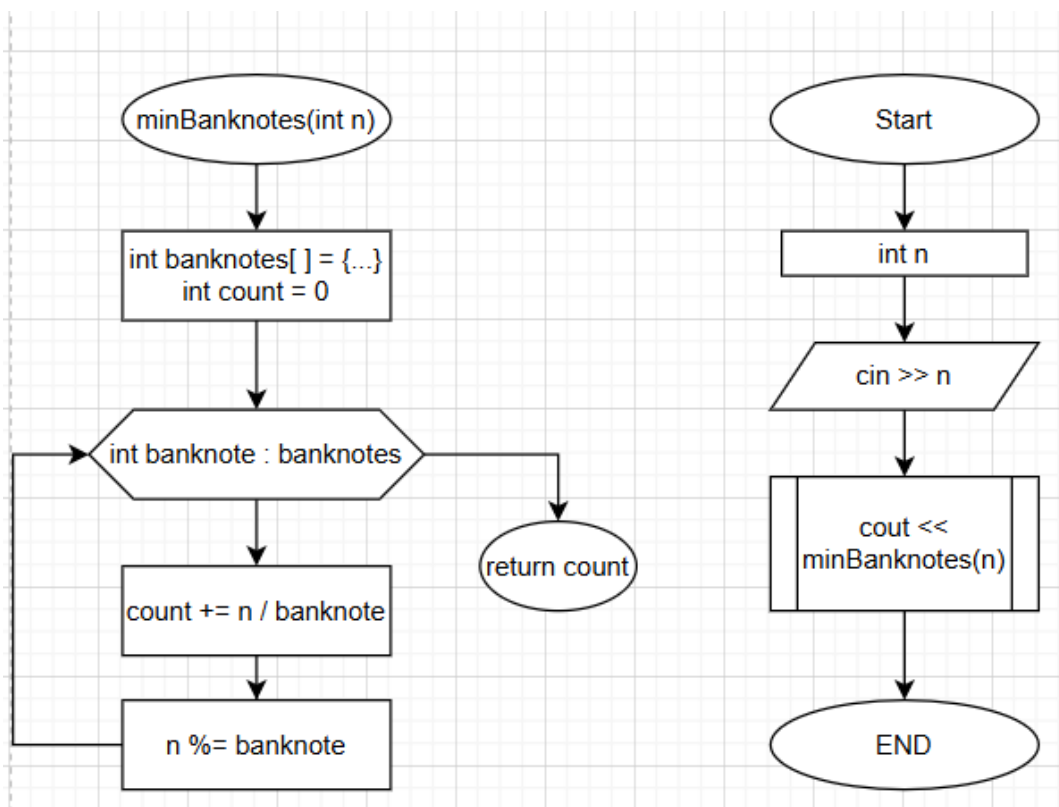
C++ 23

Accepted

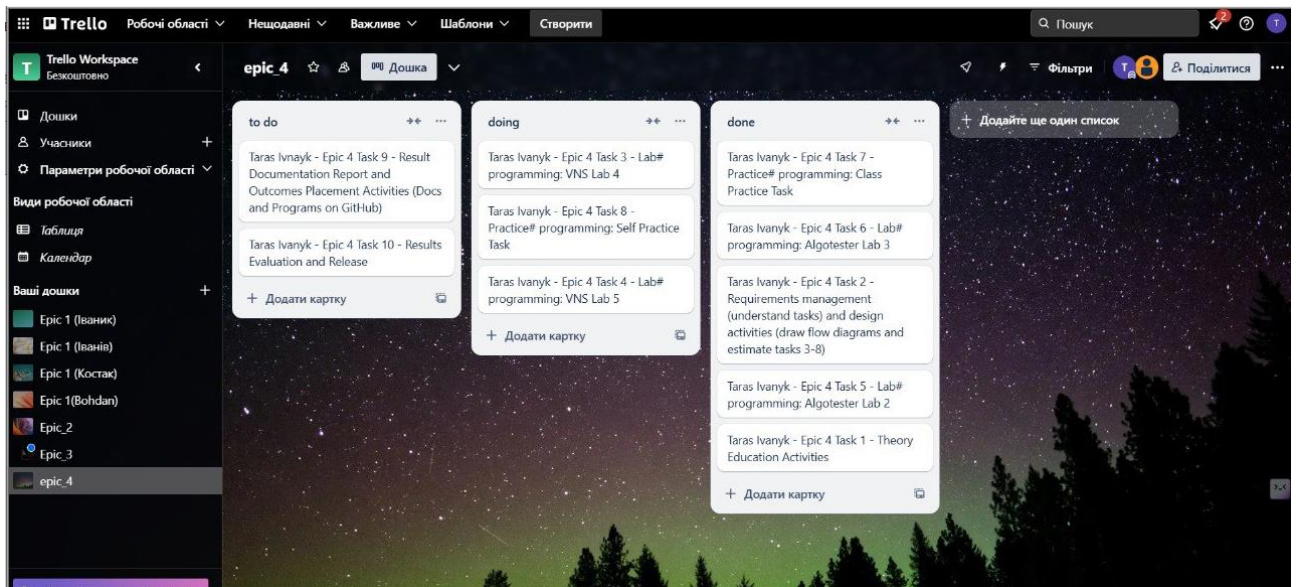
0.003

1.301

1860240



TRELLO:



Pull Request

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.