Міністерство освіти і науки України

Національний університет «Львівська політехніка»

Кафедра систем штучного інтелекту



**Звіт**

**про виконання лабораторних та практичних робіт блоку № 6**

На тему:  «Динамічні структури (Черга, Стек, Списки, Дерево). Алгоритми обробки динамічних структур.»

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

до:

ВНС Лабораторної Роботи № 10

Алготестер Лабораторної Роботи № 5

Алготестер Лабораторної Роботи № 7-8

Практичних Робіт № 6

***Виконав:***

студент групи ШІ-11

Поліщук Олександр Андрійович

# **Тема роботи:**

Динамічні структури (Черга, Стек, Списки, Дерево). Алгоритми обробки динамічних структур.

# **Мета роботи:**

Виконати практичні та лабораторні завдання, опрацювавши теорію.

# **Теоретичні відомості:**

1. Теоретичні відомості з переліком важливих тем:

* Тема №1: Зв’язні списки.
* Тема №2: Двозв’язні списки.
* Тема №3: Бінарні дерева.

1. Індивідуальний план опрацювання теорії:

* Тема №1:
  + Джерела Інформації
    - <https://www.geeksforgeeks.org/introduction-to-linked-list-data-structure-and-algorithm-tutorial/>
  + Що опрацьовано:
    - Вищевказані матеріали
  + Статус: Ознайомлений частково
  + Початок опрацювання теми: 07.12.23
  + Звершення опрацювання теми: 09.12.23
* Тема №2:
  + Джерела Інформації:
    - <https://www.geeksforgeeks.org/data-structures/linked-list/doubly-linked-list/>
  + Що опрацьовано:
    - Вищевказані матеріали
  + Статус: Ознайомлений
  + Початок опрацювання теми: 07.12.23
  + Звершення опрацювання теми: 09.12.23
* Тема №3:
  + Джерела Інформації:
    - <https://www.geeksforgeeks.org/applications-advantages-and-disadvantages-of-binary-tree/?ref=lbp>
    - <https://www.programiz.com/dsa/binary-tree>
    - <https://www.geeksforgeeks.org/tree-traversals-inorder-preorder-and-postorder/>
  + Що опрацьовано:
    - Вищевказані матеріали
  + Статус: Ознайомлений частково
  + Початок опрацювання теми: 07.12.23
  + Звершення опрацювання теми: 09.12.23

# **Виконання роботи:**

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

Завдання №1 VNS Lab 10

* Варіант завдання - 17

Завдання №2 Algotester Lab 78

* Варіант завдання - 2

Завдання №3 Algotester Lab 5

* Варіант завдання — 2

Завдання №4 Class practice task

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

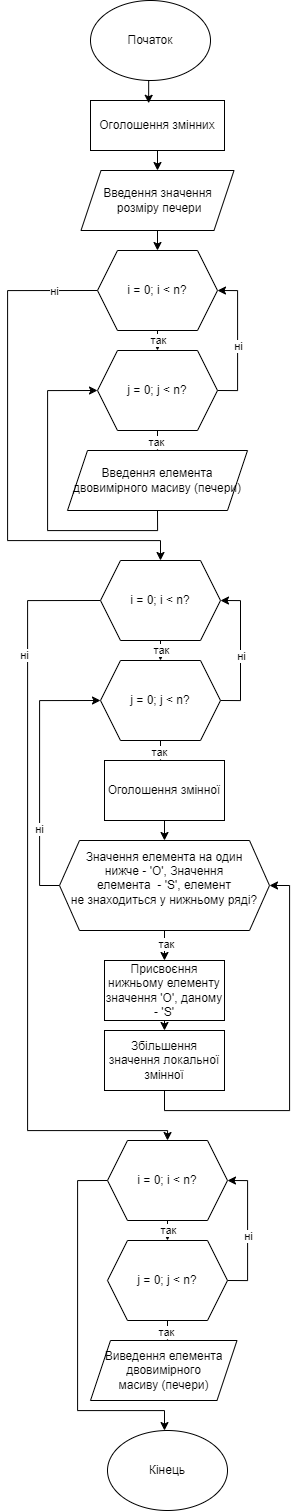
Завдання №1 VNS Lab 10

* Планований час на реалізацію — 70 хв

Завдання №2 Algotester Lab 78

* Планований час на реалізацію — 120 хв

Завдання №3 Algotester Lab 5

Блок-схема 1 до Algotester lab 5

* Планований час на реалізацію — 30 хв

Завдання №4 Class Practice Task

* Планований час на реалізацію — 4 год.

## **3. Конфігурація середовища до виконання завдань:**

Змін до середовища виконання завдань внесено не було

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

**Завдання №1 VNS Lab 10**

#include <iostream>

#include <fstream>

#include <cstring>

using namespace std;

class Node

{

public:

char\* data;

Node\* prev;

Node\* next;

public:

Node(char\* appendedData)

{

data = new char[strlen(appendedData) + 1];

strcpy(data, appendedData);

prev = NULL;

next = NULL;

}

};

class List

{

public:

Node\* head;

Node\* tail;

public:

List()

{

head = NULL;

tail = NULL;

}

Node\* push\_front(char\* pushData)

{

Node\* newNode = new Node(pushData);

newNode -> next = head;

if (head != NULL)

head -> prev = newNode;

if (tail == NULL)

tail = newNode;

head = newNode;

return newNode;

}

void printList (Node\* head)

{

if (head == NULL)

{

cout << "The list is empty" << endl;

return;

}

Node\* ptr = head -> next;

cout << head -> data << "\t";

while (ptr != NULL)

{

cout << ptr -> data << "\t";

ptr = ptr -> next;

}

}

void writeToFile(string filename)

{

ofstream outFile(filename);

if (!outFile.is\_open())

{

cout << "Opening error " << endl;

return;

}

Node\* current = head;

while (current != NULL)

{

outFile << current -> data << " ";

current = current -> next;

}

outFile.close();

}

void readFromFile(string filename)

{

ifstream inFile(filename);

if (!inFile.is\_open())

{

cout << "Opening error " << endl;

return;

}

char data[512];

while (inFile >> data)

{

push\_front(data);

}

inFile.close();

}

};

void deleteElement(List& list, int index)

{

int count = 1;

Node\* ptr = list.head;

while (ptr != NULL)

{

if(count == index)

{

if (ptr -> prev != NULL)

ptr -> prev -> next = ptr -> next;

else list.head = ptr -> next;

if (ptr ->next != NULL)

ptr -> next -> prev = ptr -> prev;

else list.tail = ptr -> prev;

delete ptr;

break;

}

count++;

ptr = ptr -> next;

}

}

void deleteList(List& list)

{

Node\* node = list.head;

Node\* toDelete = NULL;

while (node != NULL)

{

toDelete = node -> next;

delete node;

node = toDelete;

}

list.head = NULL;

list.tail = NULL;

}

int main()

{

int i, n;

char input[512];

string filename;

List newList;

cout << "Write a number of elements" << endl;

cin >> n;

cout << "Add the elements" << endl;

while(n > 0)

{

cin >> input;

newList.push\_front(input);

newList.printList(newList.head);

cout << endl;

n--;

}

newList.writeToFile("lab10.txt");

deleteList(newList);

newList.printList(newList.head);

cout << endl;

newList.readFromFile("lab10.txt");

newList.printList(newList.head);

deleteList(newList);

return 0;

}

Код №1 VNS Lab 10

[https://github.com/artificial-intelligence-department/ai\_programming\_playground/pull/877/files#diff-f2c28c9fceb3b980f22a0893e93d0cd5a20c80d859b5ee17e6b501001cb7864c](https://github.com/artificial-intelligence-department/ai_programming_playground/pull/877/files" \l "diff-f2c28c9fceb3b980f22a0893e93d0cd5a20c80d859b5ee17e6b501001cb7864c)

**Завдання №2 Algotester Lab 78**

#include <iostream>

#include <vector>

using namespace std;

template <typename T>

class Array

{

public:

T \*container;

int capacity;

int size;

Array(T value)

{

capacity = 1;

container = new T[capacity];

size = 0;

}

int expand()

{

int newCapacity = capacity \* 2;

T \*newArray = new T[newCapacity];

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

{

newArray[i] = container[i];

}

container = newArray;

capacity = newCapacity;

return capacity;

}

void insert(int index, int number)

{

size += number;

while (size >= capacity)

{

if (size >= capacity)

{

expand();

}

}

for (int i = size - 1; i > index - 1; i--)

{

container[i] = container[i - number];

}

T arr[number];

for (int j = 0; j < number; j++)

{

cin >> arr[j];

container[index + j] = arr[j];

}

}

string print()

{

string out;

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

{

out +=to\_string(container[i]);

out += " ";

}

return out;

}

void erase(int index, int n)

{

for (int i = index; i < index + n; i++)

{

container[i] = container[i + n];

}

size -= n;

}

int getCapacity()

{

return capacity;

}

int getSize()

{

return size;

}

T get(int index)

{

return container[index];

}

void set(int index, T newData)

{

container[index] = newData;

}

};

int main()

{

Array<int> initialArray(0);

int q;

vector<string> output;

cin >> q;

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

{

string input;

cin >> input;

if (input == "size")

{

output.push\_back(to\_string(initialArray.getSize()));

}

else if (input == "capacity")

{

output.push\_back(to\_string(initialArray.getCapacity()));

}

else if (input == "insert")

{

int index, n;

cin >> index >> n;

initialArray.insert(index, n);

}

else if (input == "print")

{

output.push\_back(initialArray.print());

}

else if (input == "set")

{

int index, data;

cin >> index >> data;

initialArray.set(index, data);

}

else if (input == "erase")

{

int index, n;

cin >> index >> n;

initialArray.erase(index, n);

}

else if (input == "get")

{

int index;

cin >> index;

output.push\_back(to\_string(initialArray.get(index)));

}

}

for (int i = 0; i < output.size(); i++)

{

cout << output[i] << endl;

}

return 0;

}

Код №2 до **Algotester Lab 78**

**[https://github.com/artificial-intelligence-department/ai\_programming\_playground/pull/877/files#diff-d5ed8cc2b5e60654b961f7231ec914b4e024fa04494a3e6af87a3156f7839b27](https://github.com/artificial-intelligence-department/ai_programming_playground/pull/877/files" \l "diff-d5ed8cc2b5e60654b961f7231ec914b4e024fa04494a3e6af87a3156f7839b27)**

**Завдання №3 Algotester Lab 5**

#include <iostream>

using namespace std;

int main() {

int n, m, y;

cin >> n >> m;

char board[n][m];

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

{

for (int j = 0; j < m; j++)

{

cin >> board[i][j];

}

}

for (int i = m - 1; i >= 0; i--)

{

for (int j = n - 1; j >= 0; j--)

{

int k = j;

while (board[k][i] == 'S' && k + 1 < n && board[k + 1][i] == 'O')

{

board[k + 1][i] = 'S';

board[k][i] = 'O';

k++;

}

}

}

cout << endl;

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

{

for (int j = 0; j < m; j++)

{

cout << board[i][j];

}

cout << endl;

}

return 0;

}

Код №3 до Algotester Lab 5

[https://github.com/artificial-intelligence-department/ai\_programming\_playground/pull/877/files#diff-e76d9019dff1f676818a475d48dd5ec03c49724b602c3da3855f27ebf5cb1be5](https://github.com/artificial-intelligence-department/ai_programming_playground/pull/877/files" \l "diff-e76d9019dff1f676818a475d48dd5ec03c49724b602c3da3855f27ebf5cb1be5)

Код №5 доAlgotester Lab 6

**Завдання №4 Class Practice task 1**

#include <iostream>

#include <vector>

using namespace std;

class Node

{

public:

int data;

Node\* prev;

Node\* next;

public:

Node(int appendedData)

{

data = appendedData;

prev = NULL;

next = NULL;

}

};

class List

{

public:

Node\* head;

Node\* tail;

public:

List()

{

head = NULL;

tail = NULL;

}

Node\* push\_front(int pushData)

{

Node\* newNode = new Node(pushData);

newNode -> next = head;

if (head != NULL)

head -> prev = newNode;

if (tail == NULL)

tail = newNode;

head = newNode;

return newNode;

}

Node\* push\_back(int pushData)

{

Node\* newNode = new Node(pushData);

newNode -> next = NULL;

newNode -> prev = tail;

if (head == NULL)

head = newNode;

if (tail != NULL)

tail -> next = newNode;

tail = newNode;

return newNode;

}

void printList (Node\* head)

{

if (head == NULL)

{

return;

}

Node\* ptr = head -> next;

Node\* temp = NULL;

cout << head -> data << "\t";

while (ptr != NULL)

{

cout << ptr -> data << "\t";

ptr = ptr -> next;

}

}

void reverse (Node\*& head)

{

if (head == NULL)

{

return;

}

Node\* ptr = head;

Node\* buff = NULL;

while (ptr != NULL)

{

buff = ptr -> prev;

ptr -> prev = ptr -> next;

ptr -> next = buff;

head = ptr;

ptr = ptr -> prev;

}

}

};

int main()

{

int i, n, input;

cout << "Write a number of elements" << endl;

cin >> n;

List newList;

Node\* temp = NULL;

Node\* ptr;

cout << "Add the elements" << endl;

while(n > 0)

{

cin >> input;

newList.push\_back(input);

n--;

}

newList.reverse(newList.head);

cout << "Reversed list:" << endl;

newList.printList(newList.head);

return 0;

}

Код №4 до Class Practice task 1

**Завдання №4 Class Practice task 2**

#include <iostream>

//#include <vector>

using namespace std;

class Node

{

public:

int data;

Node\* prev;

Node\* next;

public:

Node(int appendedData)

{

data = appendedData;

prev = NULL;

next = NULL;

}

};

class List

{

public:

Node\* head;

Node\* tail;

public:

List()

{

head = NULL;

tail = NULL;

}

Node\* push\_back(int pushData)

{

Node\* newNode = new Node(pushData);

newNode -> next = NULL;

newNode -> prev = tail;

if (head == NULL)

head = newNode;

if (tail != NULL)

tail -> next = newNode;

tail = newNode;

return newNode;

}

};

bool compare(List l1, List l2)

{

if (l1.head ==NULL || l2.head ==NULL)

{

return false;

}

Node\* n1 = l1.head -> next;

Node\* n2 = l2.head -> next;

int count = 1, size1 = 0, size2 = 0;

while (n1 != NULL)

{

n1 = n1 -> next;

size1++;

}

while (n2 != NULL)

{

n2 = n2 -> next;

size2++;

}

n1 = l1.head;

n2 = l2.head;

if (size1 != size2)

return false;

else

{

while (n1 != NULL && n2 != NULL)

{

if (n1 -> data > n2 -> data)

cout << "Node №" << count << "from first list is greater" << endl;

else if (n1 -> data < n2 -> data)

cout << "Node №" << count << "from second list is greater" << endl;

else cout << "Nodes №" << count << "are equal" << endl;

n1 = n1 -> next;

n2 = n2 -> next;

count++;

}

}

return true;

}

int main()

{

int input;

List l1, l2;

l1.push\_back(1);

l1.push\_back(2);

l1.push\_back(3);

l1.push\_back(4);

l2.push\_back(4);

l2.push\_back(3);

l2.push\_back(2);

l2.push\_back(1);

if (!compare(l1, l2))

cout << "Lists cannot be fully compared";

return 0;

}

Код до Class Practice task 2

[https://github.com/artificial-intelligence-department/ai\_programming\_playground/pull/877/files#diff-a0cef907dcce6e6b5ac75d11b3e594dcdadb26431feaae4169ff276e3ed3eb18](https://github.com/artificial-intelligence-department/ai_programming_playground/pull/877/files" \l "diff-a0cef907dcce6e6b5ac75d11b3e594dcdadb26431feaae4169ff276e3ed3eb18)

**Завдання №4 Class Practice task 3**

#include <iostream>

#include <cmath>

using namespace std;

class Node

{

public:

int data;

Node\* prev;

Node\* next;

public:

Node(int appendedData)

{

data = appendedData;

prev = NULL;

next = NULL;

}

};

class List

{

public:

Node\* head;

Node\* tail;

public:

List()

{

head = NULL;

tail = NULL;

}

int size()

{

int size = 0;

Node\* n = head;

while (n != NULL)

{

n = n -> next;

size++;

}

return size;

}

Node\* push\_back(int pushData)

{

Node\* newNode = new Node(pushData);

newNode -> next = NULL;

newNode -> prev = tail;

if (head == NULL)

head = newNode;

if (tail != NULL)

tail -> next = newNode;

tail = newNode;

return newNode;

}

void printList (Node\* head)

{

if (head == NULL)

{

return;

}

Node\* ptr = head -> next;

Node\* temp = NULL;

cout << head -> data << "\t";

while (ptr != NULL)

{

cout << ptr -> data << "\t";

ptr = ptr -> next;

}

}

};

List add(List l1, List l2)

{

List finalList;

int n1 = 0, n2 = 0, sum = 0, size1 = l1.size(), size2 = l2.size(), buff = 0;

Node\* ptr1 = l1.head;

Node\* ptr2 = l2.head;

buff = size1;

while (size1 != 0)

{

n1 += (ptr1 -> data) \* pow(10, (buff - size1));

ptr1 = ptr1 -> next;

size1--;

}

buff = size2;

while (size2 != 0)

{

n2 += (ptr2 -> data) \* pow(10, (buff - size2));

ptr2 = ptr2 -> next;

size2--;

}

sum = n1 + n2;

finalList.push\_back(sum);

return finalList;

}

int main()

{

List l1, l2, l3;

l1.push\_back(9);

l1.push\_back(7);

l1.push\_back(3);

l2.push\_back(4);

l2.push\_back(2);

l2.push\_back(5);

l3 = add(l1, l2);

cout << "Final list:" << endl;

l3.printList(l3.head);

return 0;

}

Код до Class Practice task 3

[https://github.com/artificial-intelligence-department/ai\_programming\_playground/pull/877/files#diff-2a5743d924032fc003766f9e8aed7e4acaddf6db831e5dd130397e45eadab77b](https://github.com/artificial-intelligence-department/ai_programming_playground/pull/877/files" \l "diff-2a5743d924032fc003766f9e8aed7e4acaddf6db831e5dd130397e45eadab77b)

**Завдання №4 Class Practice task 4**

#include <iostream>

#include <cmath>

using namespace std;

class Node

{

public:

int data;

Node\* right;

Node\* left;

public:

Node(int appendedData)

{

data = appendedData;

right = NULL;

left = NULL;

}

};

class Tree

{

public:

Node\* root;

Tree()

{

root = NULL;

}

void insert(int appendData)

{

if (root == NULL)

{

root = new Node (appendData);

return;

}

Node\* newNode = root;

while(newNode != NULL)

{

if(newNode -> data == appendData)

{

return;

}

else if (appendData > newNode -> data)

{

if(newNode -> right == NULL)

{

newNode -> right = new Node(appendData);

return;

}

else newNode = newNode -> right;

}

else if(appendData < newNode -> data)

{

if(newNode -> left == NULL)

{

newNode -> left = new Node(appendData);

return;

}

else newNode = newNode -> left;

}

}

}

void printTree(Node\* node)

{

if (node != NULL)

{

if (node->left != NULL)

{

printTree(node->left);

}

cout << node->data << "\t";

if (node->right != NULL)

{

printTree(node->right);

}

}

}

};

Node\* create\_mirror\_flip(Node\* root)

{

Node\* node = root;

if(node != NULL)

{

Node\* buff = node -> left;

node -> left = node -> right;

node -> right = buff;

create\_mirror\_flip(node -> left);

create\_mirror\_flip(node -> right);

}

return node;

}

Tree create\_mirror\_flip(Tree tree)

{

tree.root = create\_mirror\_flip(tree.root);

return tree;

}

int main()

{

Tree newTree;

Tree mirrorTree;

newTree.insert(8);

newTree.insert(6);

newTree.insert(10);

newTree.insert(2);

newTree.insert(11);

newTree.insert(9);

newTree.insert(7);

newTree.printTree(newTree.root);

mirrorTree = create\_mirror\_flip(newTree);

cout << "\n" << "Mirror tree: " << endl;

newTree.printTree(mirrorTree.root);

return 0;

}

Код до Class Practice task 4

[https://github.com/artificial-intelligence-department/ai\_programming\_playground/pull/877/files#diff-9fdd4f1b7a672254c3a37b150158dee14fd6902b6d326f74e5c4141981d6f58c](https://github.com/artificial-intelligence-department/ai_programming_playground/pull/877/files" \l "diff-9fdd4f1b7a672254c3a37b150158dee14fd6902b6d326f74e5c4141981d6f58c)

**Завдання №4 Class Practice task 5**

#include <iostream>

using namespace std;

class Node

{

public:

int data;

Node\* right;

Node\* left;

public:

Node(int appendedData)

{

data = appendedData;

right = NULL;

left = NULL;

}

void insert(int appendData)

{

Node\* newNode = this;

while(newNode != NULL)

{

if(newNode -> data == appendData)

{

return;

}

else if (appendData > newNode -> data)

{

if(newNode -> right == NULL)

{

newNode -> right = new Node(appendData);

return;

}

else newNode = newNode -> right;

}

else if(appendData < newNode -> data)

{

if(newNode -> left == NULL)

{

newNode -> left = new Node(appendData);

return;

}

else newNode = newNode -> left;

}

}

}

};

int sum(Node\* node)

{

int s = 0;

if (node -> left != NULL)

s += sum(node -> left);

if (node -> right != NULL)

s += sum(node -> right);

if(node -> left == NULL && node -> right == NULL)

s = node -> data;

node -> data = s;

return s;

}

void printTree(Node\* node)

{

if (node != NULL)

{

if (node->left != NULL)

{

printTree(node->left);

}

cout << node->data << "\t";

if (node->right != NULL)

{

printTree(node->right);

}

}

}

Node\* forNode (Node\* node)

{

if (node -> left != NULL)

node = node -> left;

if (node -> right != NULL)

node = node -> right;

node -> data = sum(node);

return node;

}

int main()

{

int a;

Node\* newTree = new Node(8);

newTree -> insert(6);

newTree -> insert(10);

newTree -> insert(2);

newTree -> insert(11);

newTree -> insert(9);

newTree -> insert(7);

printTree(newTree);

sum(newTree);

cout << "\n" << "Sum :" << endl;

printTree(newTree);

return 0;

}

Код до Class Practice task 5

[https://github.com/artificial-intelligence-department/ai\_programming\_playground/pull/877/files#diff-339ecee653b3071a19aa3912a63d8bfe42e6864c377ccc735dbf66fc20f6f0c6](https://github.com/artificial-intelligence-department/ai_programming_playground/pull/877/files" \l "diff-339ecee653b3071a19aa3912a63d8bfe42e6864c377ccc735dbf66fc20f6f0c6)

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

Завдання №1 VNS Lab 10

PPS C:\Users\alexs> & 'c:\Users\alexs\.vscode\extensions\ms-vscode.cpptools-1.18.5-win32-x64\debugAdapters\bin\WindowsDebugLauncher.exe' '--stdin=Microsoft-MIEngine-In-y1j3nd2a.x3h' '--stdout=Microsoft-MIEngine-Out-1fmhhaoq.oth' '--stderr=Microsoft-MIEngine-Error-hv0ilkoq.vg2' '--pid=Microsoft-MIEngine-Pid-xrimfmcj.j21' '--dbgExe=C:\msys64\ucrt64\bin\gdb.exe' '--interpreter=mi'

Write a number of elements

5

Add the elements

The

The

perfect

perfect The

doubly

doubly perfect The

linked

linked doubly perfect The

list

list linked doubly perfect The

The list is empty

The perfect doubly linked list

Результат виконання VNS Lab 10

Час затрачений на виконання завдання — 1 год

Завдання №2 Algotester Lab 78

PS C:\Users\alexs> & 'c:\Users\alexs\.vscode\extensions\ms-vscode.cpptools-1.18.5-win32-x64\debugAdapters\bin\WindowsDebugLauncher.exe' '--stdin=Microsoft-MIEngine-In-quyvmyt2.m1n' '--stdout=Microsoft-MIEngine-Out-vsnyzjfi.0st' '--stderr=Microsoft-MIEngine-Error-rjif5uc0.x1d' '--pid=Microsoft-MIEngine-Pid-qvz3tw4c.zht' '--dbgExe=C:\msys64\ucrt64\bin\gdb.exe' '--interpreter=mi'

12

size

insert 0 5

251 252 253 254 255

size

capacity

print

get 1

set 1 777

get 1

erase 1 3

get 1

size

print

0

5

8

251 252 253 254 255

252

777

255

2

251 255

Результат виконання Algotester Lab 78

Час затрачений на виконання завдання - 5 год

Завдання №3 Algotester Lab 5

PS C:\Users\alexs> & 'c:\Users\alexs\.vscode\extensions\ms-vscode.cpptools-1.18.5-win32-x64\debugAdapters\bin\WindowsDebugLauncher.exe' '--stdin=Microsoft-MIEngine-In-5xh3x0py.x3u' '--stdout=Microsoft-MIEngine-Out-gkbuu4yg.g31' '--stderr=Microsoft-MIEngine-Error-4uf02i1a.dva' '--pid=Microsoft-MIEngine-Pid-p3ahl2k0.kcf' '--dbgExe=C:\msys64\ucrt64\bin\gdb.exe' '--interpreter=mi'

5 5

SSOSS

OOOOO

SOOXX

OOOOS

OOSOO

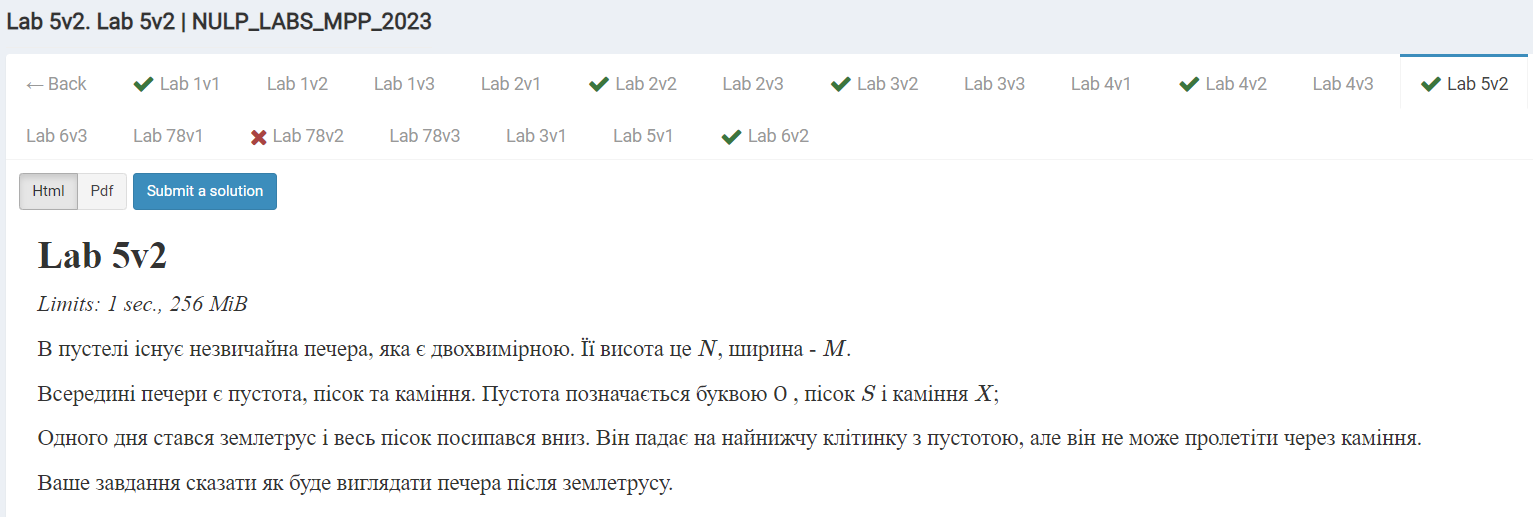
OOOOO

OOOSS

OOOXX

SOOOO

SSSOS

Виконане завдання на Algotester

Результат виконання Algotester Lab 5

Час затрачений на виконання завдання - 3 год

Завдання №4 Class Practice task 1

PS C:\Users\alexs> & 'c:\Users\alexs\.vscode\extensions\ms-vscode.cpptools-1.18.5-win32-x64\debugAdapters\bin\WindowsDebugLauncher.exe' '--stdin=Microsoft-MIEngine-In-d2tpxdvj.g3g' '--stdout=Microsoft-MIEngine-Out-wquzrzml.5o1' '--stderr=Microsoft-MIEngine-Error-satiw4wa.ddw' '--pid=Microsoft-MIEngine-Pid-ljoblt2z.1xn' '--dbgExe=C:\msys64\ucrt64\bin\gdb.exe' '--interpreter=mi'

Write a number of elements

10

Add the elements

2 5 1 7 34 4 9 0 3 10

Reversed list:

10 3 0 9 4 34 7 1 5 2

Результат виконання Class Practice task 1

Час затрачений на виконання завдання — 2 год

Завдання №4 Class Practice task 2

PS C:\Users\alexs> & 'c:\Users\alexs\.vscode\extensions\ms-vscode.cpptools-1.18.5-win32-x64\debugAdapters\bin\WindowsDebugLauncher.exe' '--stdin=Microsoft-MIEngine-In-2mil2slc.s1g' '--stdout=Microsoft-MIEngine-Out-hk55tame.usw' '--stderr=Microsoft-MIEngine-Error-hno4nwup.u2h' '--pid=Microsoft-MIEngine-Pid-b4n13xre.sdq' '--dbgExe=C:\msys64\ucrt64\bin\gdb.exe' '--interpreter=mi'

Node №1from second list is greater

Node №2from second list is greater

Node №3from first list is greater

Node №4from first list is greater

Результат виконання Class Practice task 2

Час затрачений на виконання завдання — 30хв

Завдання №4 Class Practice task 3

PS C:\Users\alexs> & 'c:\Users\alexs\.vscode\extensions\ms-vscode.cpptools-1.18.5-win32-x64\debugAdapters\bin\WindowsDebugLauncher.exe' '--stdin=Microsoft-MIEngine-In-msgnr2eh.esm' '--stdout=Microsoft-MIEngine-Out-oukg31ot.tum' '--stderr=Microsoft-MIEngine-Error-iloaqaau.4wo' '--pid=Microsoft-MIEngine-Pid-45rnafzp.ep2' '--dbgExe=C:\msys64\ucrt64\bin\gdb.exe' '--interpreter=mi'

Final list:

903

Результат виконання Class Practice task 3

Час затрачений на виконання завдання — 30хв

Завдання №4 Class Practice task 4

PS C:\Users\alexs> & 'c:\Users\alexs\.vscode\extensions\ms-vscode.cpptools-1.18.5-win32-x64\debugAdapters\bin\WindowsDebugLauncher.exe' '--stdin=Microsoft-MIEngine-In-aa3a5ysi.14a' '--stdout=Microsoft-MIEngine-Out-pkc4lyur.q0u' '--stderr=Microsoft-MIEngine-Error-g5phbvvn.hd5' '--pid=Microsoft-MIEngine-Pid-qfgrek4a.ltu' '--dbgExe=C:\msys64\ucrt64\bin\gdb.exe' '--interpreter=mi'

2 6 7 8 9 10 11

Mirror tree:

11 10 9 8 7 6 2

Результат виконання Class Practice task 4

Час затрачений на виконання завдання — 2 год

Завдання №4 Class Practice task 5

PS C:\Users\alexs> & 'c:\Users\alexs\.vscode\extensions\ms-vscode.cpptools-1.18.5-win32-x64\debugAdapters\bin\WindowsDebugLauncher.exe' '--stdin=Microsoft-MIEngine-In-pdgqnqxz.wt0' '--stdout=Microsoft-MIEngine-Out-50gyoby2.mh3' '--stderr=Microsoft-MIEngine-Error-qhheoyfr.vt2' '--pid=Microsoft-MIEngine-Pid-glyrofcd.rxo' '--dbgExe=C:\msys64\ucrt64\bin\gdb.exe' '--interpreter=mi'

2 6 7 8 9 10 11

Sum :

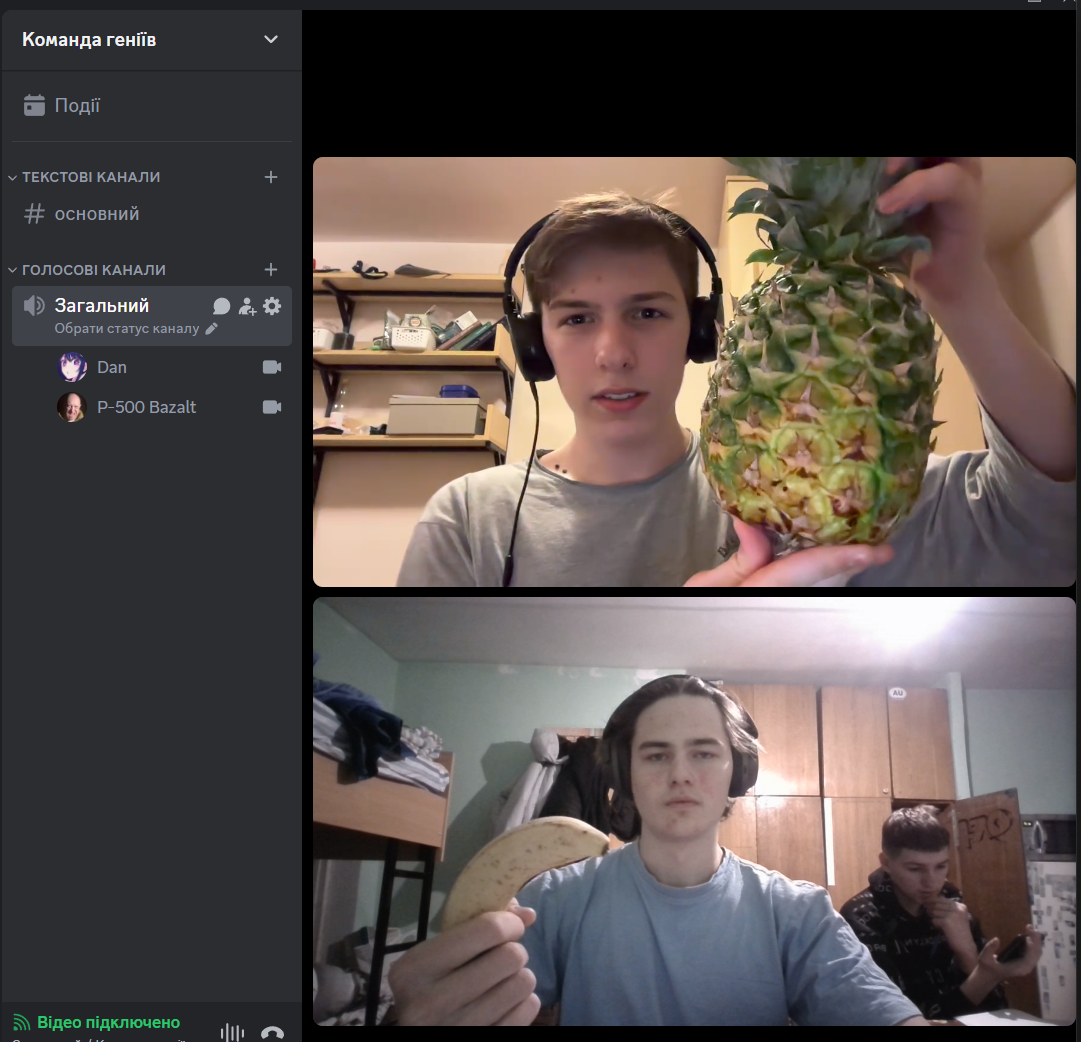
2 9 7 29 9 20 11

Результат виконання Class Practice task 5

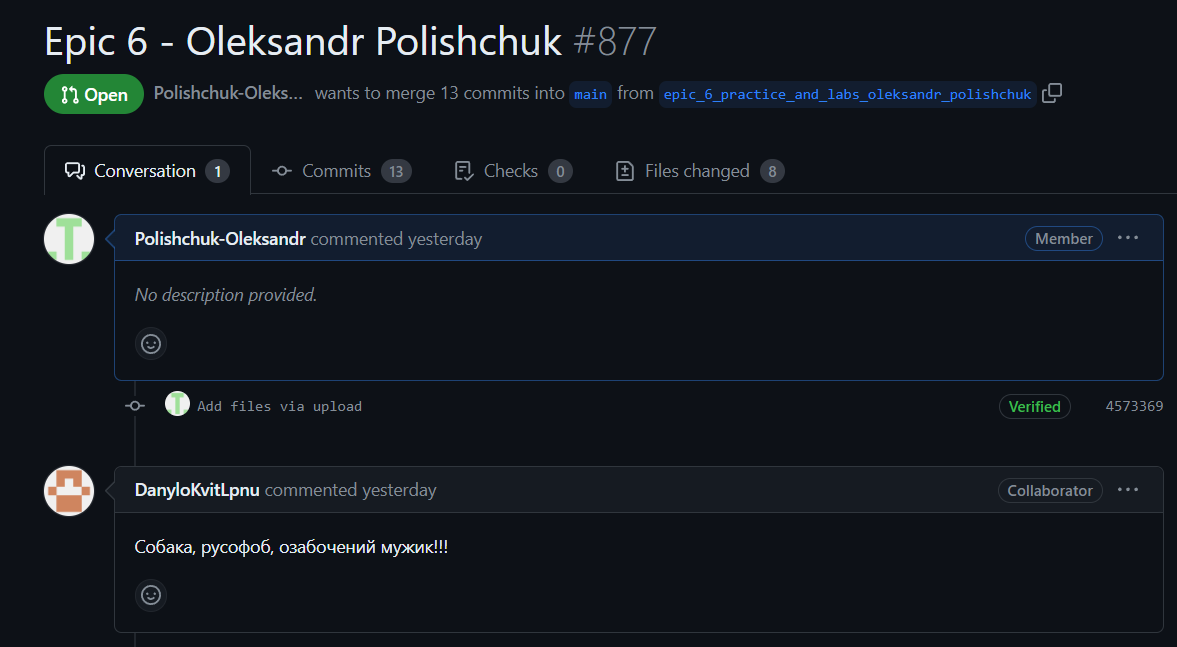
Час затрачений на виконання завдання — 2 год

## **6. Командна робота:**

Ілюстрація 1: Обговорення завдань у онлайн-режимі

Ілюстрація 1: Обговорення завданнь у онлайн режимі

# **+**

Ілюстрація 2: Коментар товариша по команді до завдання

# **Висновки:**

Лабораторні та практичні роботи виконано, опрацьовано новий матеріал (робота зі стрічками та файлами).