НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ

«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ

імені ІГОРЯ СІКОРСЬКОГО»

ЗВІТ

З лабораторної роботи №5

з навчальної дисципліни

«Програмцвання алгоритмічниї структур»

Варіант № 7

Тема:

**Методи пошуку у масивах**

Студентки 1-го курсу НН ІАТЕ гр. ТР-22

Грищук Анжеліка

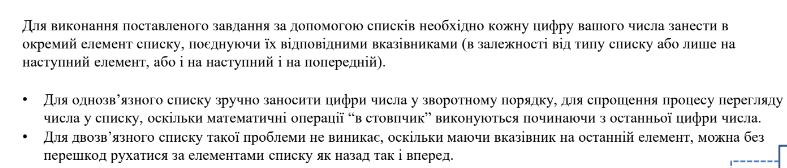
Лабораторну роботу захищено

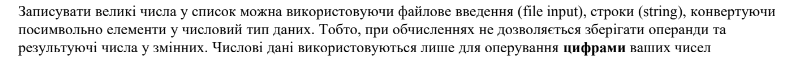
З оцінкою\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

КИЇВ 2023

**Мета роботи:** Метою лабораторної роботи є ознайомлення з основами створення та особливостями роботи однозв’язних тадвозв’язних списків, набуття навичок застосування структур даних даного виду для вирішення практичних задач.

**Завдання до лабораторної роботи:**







**Код програми:**

#include <iostream>

#include <fstream>

#include <string>

#include <windows.h>

using namespace std;

struct Node {

int data;

Node\* next;

Node\* prev;

Node(int value) : data(value), next(nullptr), prev(nullptr) {}

};

class DoublyLinkedList {

private:

Node\* head;

Node\* tail;

public:

DoublyLinkedList() : head(nullptr), tail(nullptr) {}

void insertFront(int value) {

Node\* newNode = new Node(value);

if (head == nullptr) {

head = newNode;

tail = newNode;

}

else {

newNode->next = head;

head->prev = newNode;

head = newNode;

}

}

void insertBack(int value) {

Node\* newNode = new Node(value);

if (tail == nullptr) {

head = newNode;

tail = newNode;

}

else {

newNode->prev = tail;

tail->next = newNode;

tail = newNode;

}

}

void multiply(DoublyLinkedList& other) {

DoublyLinkedList result;

Node\* currOtherDigit = other.tail;

int otherDigit = currOtherDigit->data;

int digitMultiplier = 1;

while (currOtherDigit != nullptr) {

Node\* currDigit = tail;

int carry = 0;

while (currDigit != nullptr) {

int product = currDigit->data \* otherDigit + carry;

int digit = product % 10;

carry = product / 10;

result.insertFront(digit);

currDigit = currDigit->prev;

}

if (carry > 0) {

result.insertFront(carry);

}

int zerosToAdd = digitMultiplier - 1;

while (zerosToAdd > 0) {

result.insertBack(0);

zerosToAdd--;

}

currOtherDigit = currOtherDigit->prev;

if (currOtherDigit != nullptr) {

otherDigit = currOtherDigit->data;

digitMultiplier \*= 10;

}

}

// Print the result

cout << " |--> ";

Node\* currNode = result.head;

while (currNode != nullptr) {

cout << currNode->data;

currNode = currNode->next;

}

cout << " |" << endl;

}

};

int main() {

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

cout << " |--> 18446744073709551615 |" << endl;

cout << "\* |" << "\t\t\t " << "|" << endl;

cout << " |--> 5" << "\t\t " << "|" << endl;

cout << "--|-------------------------|" << endl;

ifstream inputFile("input.txt");

if (!inputFile.is\_open()) {

cout << "Невдалося відкрити файл!" << endl;

return 0;

}

string expression;

getline(inputFile, expression);

inputFile.close();

size\_t separatorPos = expression.find('\*');

if (separatorPos == string::npos) {

cout << "Недійсний формат введення. Потрібно 'number \* number'." << endl;

return 0;

}

string num1Str = expression.substr(0, separatorPos);

string num2Str = expression.substr(separatorPos + 1);

DoublyLinkedList num1;

DoublyLinkedList num2;

for (char c : num1Str) {

if (isdigit(c)) {

int digit = c - '0';

num1.insertBack(digit);

}

}

for (char c : num2Str) {

if (isdigit(c)) {

int digit = c - '0';

num2.insertBack(digit);

}

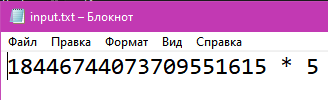
}

num1.multiply(num2);

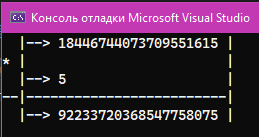
return 0;

}

**дані що заносяться в списки:**



**Результат обчислення:**



**Висновки:**

У результаті виконання даної лабораторної роботи було я ознайомилася з основами створення та особливостями роботи однозв'язних та двозв'язних списків. Набула необхідних навичок для застосування цих структур даних у практичних задачах.