Міністерство освіти і науки України
Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського"
Факультет інформатики та обчислювальної техніки

Кафедра автоматизованих систем обробки інформації і управління

Звіт

з лабораторної роботи № 4 з дисципліни
«Алгоритми та структури даних 2. Структури даних»

«Спискові структури даних»

Виконав(ла) *ІП-02 Гущін Д.О.* _____
(шифр, прізвище, ім'я, по батькові)

Перевірив *Головченко М.М.* _____
(прізвище, ім'я, по батькові)

Київ 2021

ЗМІСТ

1 МЕТА ЛАБОРАТОРНОЇ РОБОТИ

Мета роботи – розглянути базові структури даних і методи вирішення типових задач з їх допомогою.

2 ЗАВДАННЯ

Розробити алгоритм розв’язання задачі відповідно до варіанту. Виконати програмну реалізацію задачі. Не використовувати вбудовані спискові структури даних (контейнери). Зробити висновок по лабораторній роботі.

Варіанти завдань.

1. Заданий текст, що має декілька рядків. Використовуючи стек, елементами якого є літери, надрукувати текст, в якому літери кожного рядка містяться у зворотному порядку.

2. Побудувати список, елементами якого є дійсні числа. Знайти їх середнє арифметичне і розмістити отримане значення останнім елементом списку. Надрукувати початковий і змінений списки.

3. Заданий рядок слів, які відокремлюються одне від одного пробілами. Побудувати список, елементами якого є відповідні слова. Вилучити із списку всі слова, що починаються та закінчуються на задану користувачем літеру. Надрукувати початковий і змінений списки.

4. Задане натуральне число n та послідовність дійсних чисел $x_1, ..., x_n$. Використовуючи двозв'язний список, елементами якого є задані дійсні числа, визначити $(x_1 - x_n) + (x_2 - x_{n-1}) + ... + (x_n - x_1)$.

5. Заданий текст, що містить декілька рядків слів, розділених пробілами. Використовуючи стек, елементами якого є слова, надрукувати текст, в якому слова кожного рядка містяться у зворотному порядку.

6. Заданий рядок слів, які відокремлюються одне від одного комами. Побудувати список, елементами якого є відповідні слова. Вилучити із списку всі слова заданої користувачем довжини. Надрукувати початковий і змінений списки.

7. Побудувати список, елементами якого є цілі числа. Знайти суму останнього і передостаннього його елементів і замістити відповідним значенням ці елементи списку. Надрукувати початковий і змінений списки.

8. Заданий текст, що має декілька рядків. Використовуючи чергу або стек,

елементами яких є літери, надрукувати тексти, що знаходяться між кожною парою дужок заданого користувачем виду.

9. Заданий рядок слів, які відокремлюються одне від одного будь-якою кількістю пробілів. Побудувати список відповідних слів. Поміняти місцями найдовше та найкоротше слово цього списку. Надрукувати початковий і змінений списки.

10. Одне з можливих представлень тексту — це розділити його на рядки і створити список рядків, додавши ознаку кінця тексту. Використовуючи дане представлення тексту, визначити, чи входить задана літера у текст, і, якщо входить, то вивести «координати» першого входження цієї літери у текст (номер рядка і номер позиції в цьому рядку).

11. Заданий рядок слів, які відокремлюються одне від одного символами ";". Побудувати список слів, що містяться у цьому рядку. Вилучити із списку всі однакові слова. Надрукувати початковий і змінений списки.

12. Побудувати список символів. Визначити найбільше значення списку і помістити його на початок списку. Надрукувати початковий і змінений списки.

13. Задане натуральне число n та послідовність дійсних чисел $x_1, ..., x_n$. Використовуючи двозв'язний список, елементами якого є задані дійсні числа, визначити $x_1 \cdot x_n + x_2 \cdot x_{n-1} + ... + x_n \cdot x_1$.

14. Заданий список символів. Перенести в кінець цього списку його перший елемент. Надрукувати початковий та змінений списки.

15. Задана послідовність натуральних чисел. Використовуючи стек, надрукувати у зворотному порядку усі числа, які містяться між найбільшим та найменшим числами послідовності.

16. Текстовий файл містить вираз, записаний у звичайній (інфіксній) формі. Використавши стек, перекласти заданий вираз в постфіксну форму і записати в новий текстовий файл. Інфіксна форма виразу: a-b, a*b; постфіксна форма: ab-, ab+.

17. Задана послідовність цілих чисел. Використовуючи стек, надрукувати у зворотному порядку усі її числа, що не кратні 5.

18. Створити кільцевий список, елементами якого є числа. Послідовно вилучати кожне третє число. Підрахувати кількість вилучень. Вивести початковий список та вилучені елементи у порядку їх вилучення.

19. Задана послідовність дійсних чисел, що містить від'ємні елементи. Побудувати список, елементами якого є відповідні числа. Вилучити всі від'ємні елементи цього списку. Надрукувати початковий та новий списки.

20. Задана послідовність цілих чисел, що містить від'ємні елементи. Використовуючи стек, елементами якого є цілі числа, надрукувати у зворотному порядку всі додатні елементи послідовності.

21. Задане натуральне число n та послідовність дійсних чисел $x_1, ..., x_n$. Використовуючи двозв'язний список, елементами якого є задані дійсні числа, побудувати список, що містить елементи $x_1 \cdot x_n, x_2 \cdot x_n, ..., x_{n-1} \cdot x_n$.

22. Створити список цілих чисел. Вилучити із списку усі парні числа, підрахувавши їх кількість. Надрукувати початковий, змінений список та визначену величину.

23. Задана послідовність цілих чисел. Використовуючи чергу, елементами якої є цілі числа, вивести на друк спочатку парні елементи послідовності, а потім – непарні.

24. Побудувати кільцевий список, елементами якого є цілі числа. Послідовно вилучати кожне його парне число, заносячи його до стеку. Вивести початковий список та вміст стеку.

25. Створити двозв'язний список, елементами якого є слова тексту. Вивести слова, які стоять на парних місцях при проході по списку в одному напрямку, та слова, які стоять на непарних позиціях при проході по списку в зворотньому напрямку.

26. Задана послідовність цілих чисел. Побудувати список, у якому числа впорядковані у порядку зростання. Надрукувати послідовність і впорядкований список.

27. Заданий текст, що містить декілька рядків слів, розділених пробілами. Використовуючи чергу, елементами якої є слова, та стек, елементами якого є

літери, надрукувати текст, в якому літери слів кожного рядка містяться у зворотному порядку.

28. Одне з можливих представлень тексту — це розділити його на рядки і створити список рядків, додавши ознаку кінця тексту. Використовуючи дане представлення тексту, визначити номер рядка з максимальною кількістю входжень заданої літери.

29. Задане натуральне число n та послідовність дійсних чисел $x_1, ..., x_n$. Використовуючи двозв'язний список, елементами якого є задані дійсні числа, побудувати список, що містить елементи $x_1 - x_n, x_2 - x_n, ..., x_{n-1} - x_n$.

30. Заданий список, елементами якого символи. Вставити в кінець списку новий елемент, що введений з клавіатури, та вилучити із списку перший його елемент. Надрукувати початковий та змінений списки.

31. Реалізуйте структуру "черга з пріоритетом", яка підтримує наступні операції: додавання елемента в чергу; видалення з черги елемента з найбільшим пріоритетом; зміна пріоритету для довільного елемента, що знаходиться в черзі.

32. Створіть двозв'язний список груп факультету інформатики. Кожна група представляє собою однозв'язний список студентів. Надрукувати дані двозв'язного списку.

33. Створити двозв'язний список, елементами якого є цілі числа. Упорядкувати елементи списку двома способами (зміна покажчиків, зміна значень елементів).

34. Створити двозв'язний список елементами якого є цілі числа, видалити зі списку елементи, значення яких вже зустрічалися раніше. Надрукувати початковий та змінений списки.

35. Створити двозв'язний список із цілих значень та визначити чи можна видалити зі списку якихось два елементи так, щоб новий список виявився упорядкованим.

3 ВИКОНАННЯ

3.1 Псевдокод алгоритмів

```
procedure printRange(stack, max, min)
    flag = true
    while flag AND !stack.empty()
        if stack.top() == max
            while stack.top() != min
                print stack.top()
                stack.pop()
            print stack.top()
            flag = false
            end while
        end if
        else if stack.top() == min
            while stack.top() != max
                print stack.top()
                stack.pop();
            print stack.top()
            flag = false
            end while
        end else if
        stack.pop()
    end while
end procedure

function findMin(stack)
    min = INT_MAX
    while !stack.empty()
        if stack.top() < min
            min = stack.top()
        end if
        stack.pop()
    end while
    return min
end function

function findMax(stack)
    max = INT_MIN
    while !stack.empty()
        if stack.top() > max
            max = stack.top()
        end if
        stack.pop()
    end while
    return max
end function
```

3.2 Програмна реалізація

3.2.1 Вихідний код

```
#include <iostream>

using namespace std;

class Stack {
private:
    int* arr;
    int capacity;
    int size;
public:
    Stack();
    Stack(int size);
    Stack(const Stack& other);
    ~Stack();
    void push(int x);
    void pop();
    bool empty();
    int top();
};

void printRange(Stack stack, int max, int min);
int findMin(Stack stack);
int findMax(Stack stack);

int main()
{
    int size;
    cout << "Enter amount of numbers: "; cin >> size;
    Stack S;
    for (int i = 0; i < size; i++) {
        int num;
        cout << "Enter number: "; cin >> num;
        S.push(num);
    }
    int max = findMax(S);
    int min = findMin(S);
    printRange(S, max, min);
    system("pause");
    return 0;
}


void printRange(Stack stack, int max, int min) {
    bool flag = true;
    while (flag && !stack.empty()) {
        if (stack.top() == max)
        {
            cout << endl;
            while (stack.top() != min) {
                cout << stack.top() << '\t';
                stack.pop();
            }
            cout << stack.top();
            flag = false;
        }
        else if (stack.top() == min)
        {
            cout << endl;
            while (stack.top() != max) {
```

```
                cout << stack.top() << '\t';
                stack.pop();
            }
            cout << stack.top();
            flag = false;
        }
        stack.pop();
    }
    cout << endl;
}

int findMin(Stack stack) {
    int min = INT_MAX;
    while (!stack.empty()) {
        if (stack.top() < min) min = stack.top();
        stack.pop();
    }
    return min;
}

int findMax(Stack stack) {
    int max = INT_MIN;
    while (!stack.empty()) {
        if (stack.top() > max) max = stack.top();
        stack.pop();
    }
    return max;
}

Stack::Stack() { // конструктор
    size = 10;
    arr = new int[size];
    capacity = -1;
}

Stack::Stack(int size) { // конструктор
    this->size = size;
    arr = new int[size];
    capacity = -1;
}

Stack::Stack(const Stack& other) { // конструктор копирования
    this->size = other.size;
    this->capacity = other.capacity;
    this->arr = new int[size];
    if (capacity >= 0) {
        for (int i = 0; i < other.capacity; i++) {
            this->arr[i] = other.arr[i];
        }
    }
}

Stack::~Stack() { // деструктор
    delete[] arr;
}

void Stack::push(int value) {
    if (capacity == size - 1) {
        int* new_arr = new int[size * 2 + 1];
        for (int i = 0; i < size; i++) {
            new_arr[i] = arr[i];
        }
        delete[] arr;
        this->arr = new_arr;
```

```
        for (int i = 0; i < size; i++) {
            arr[i] = new_arr[i];
        }
        size = size * 2 + 1;
    }
    capacity++;
    arr[capacity] = value;
}

void Stack::pop() {
    if (capacity >= 0) {
        //arr[capacity] = 0;
        --capacity;
    }
    else {
        cout << "It's empty";
    }
}

bool Stack::empty() {
    return (capacity == 0);
}

int Stack::top() {
    return arr[capacity - 1];
}
```

3.2.2 Приклади роботи

На рисунках 3.1 і 3.2 показані приклади роботи програми.

```
Enter amount of numbers: 5
Enter number: 2
Enter number: 1
Enter number: 3
Enter number: 5
Enter number: 4

5       3       1
Для продолжения нажмите любую клавишу . . .
```

Рисунок 3.1 –

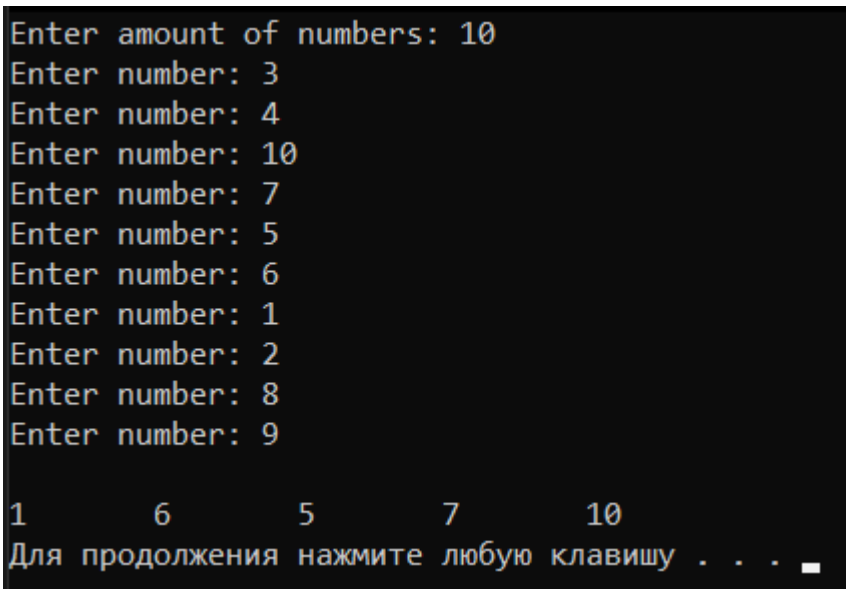
Enter amount of numbers: 10
Enter number: 3
Enter number: 4
Enter number: 10
Enter number: 7
Enter number: 5
Enter number: 6
Enter number: 1
Enter number: 2
Enter number: 8
Enter number: 9
1 6 5 7 10
Для продолжения нажмите любую клавишу . . .

Рисунок 3.2 –

ВИСНОВОК

При виконанні даної лабораторної роботи ми розглянули базові структури даних, а також розробили алгоритм розв’язання задачі відповідно до варіанту і виконали програмну реалізацію задачі на мові C++, не використовуючи вбудовані спискові структури даних (контейнери). Можна зробити висновок, що структури даних допомагають упорядкувати дані, що спрощує роботу з великим об'ємом інформації, і робить програму ефективніше.

КРИТЕРІЇ ОЦІНЮВАННЯ

За умови здачі лабораторної роботи до 20.04.2020 включно максимальний бал дорівнює – 5. Після 20.04.2020 максимальний бал дорівнює – 1.

Критерії оцінювання у відсотках від максимального балу:

- псевдокод алгоритму – 10%;
- програмна реалізація алгоритму – 80%;
- висновок – 10%.