Державний університет «Одеська політехніка»

Інститут комп'ютерних систем

Кафедра інформаційних систем

**КУРСОВА РОБОТА**

з дисципліни «Алгоритмізація та програмування»

Тема «Програмування динамічної структури даних – двобічна черга»

Студента (ки) \_\_1\_\_ курсу АІ-215 групи

Спеціальності 122 – «Комп’ютерні науки»

\_\_\_\_\_\_\_\_\_ Лисишин Д.О.\_\_\_\_\_\_\_\_\_\_\_

(прізвище та ініціали)

Керівник асист. Іванов О.В.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(посада, вчене звання, науковий ступінь, прізвище та ініціали)

Національна шкала \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Кількість балів: \_\_\_\_\_\_\_\_\_\_

Оцінка: ECTS \_\_\_\_\_

Члени комісії \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(підпис) (прізвище та ініціали)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(підпис) (прізвище та ініціали)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(підпис) (прізвище та ініціали)

м. Одеса – 2022 рік

Державний університет «Одеська політехніка»

Інститут комп'ютерних систем

Кафедра інформаційних систем

**ЗАВДАННЯ**

НА КУРСОВУ РОБОТУ

студенту Лисишину Данилу Олеговичу група АІ-215

1. Тема роботи

«Програмування динамічної структури даних – двобічна черга»

2. Термін здачі студентом закінченої роботи 3.06.2022

3. Початкові дані до проекту (роботи)

Варіант 11

Предметна область – дискотека. Реалізувати динамічну структуру даних (двобічна черга), що містить наступну інформацію: назва диска; рік випуску; тривалість звучання; виконавець. Програма повинна забезпечувати: − додавання елемента; − видалення елемента; − можливість коригування даних; − виведення всіх даних; − виведення списку елементів з роком випуску пізніше, ніж введений; − сортування по полю «рік випуску»; − пошук найтривалішого за звучанням альбому; − виведення всіх альбомів, заданого виконавця.

4. Зміст розрахунково-пояснювальної записки (перелік питань, які належить розробити)

Вступ. Теоретичні відомості про двобічну чергу. Програмна реалізація двобічної черги. Інструкція користувача. Висновки. Перелік використаних джерел. Додатки

5. Перелік графічного матеріалу (з точним зазначенням обов’язкових креслень)

Блок-схема алгоритму – 1 аркуш формату А1.

Завдання видано 28.03.22 \_\_\_\_\_\_\_\_\_\_\_\_\_\_

(підпис викладача)

Завдання прийнято до виконання 28.03.22 \_\_\_\_\_\_\_\_\_\_\_\_\_\_

(підпис студента)

**АНОТАЦІЯ**

Розглянуто динамічну структуру данних двобічна черга. Розглянута реалізація динамічної структури данних двобчна черга на мові С.Розроблена та запропонована до перегляду блок-схема всього програмного коду.Запропонована інструкція користувача, у якій демонструється весь доступний функціонал.Запропонована своя реалізація двобічної черги та програмних функцій.

**ABSTRACT**

The dynamic data structure of the two-way queue is considered. The implementation of the dynamic data structure of the two-sided queue in the C language is considered. A block diagram of all program code has been developed and offered for viewing. A user manual is provided, which demonstrates all available functionality. he implementation of the bilateral queue and program functions is offered.

**ЗМІСТ**

[ВСТУП 7](#_Toc104925569)

[1 ТЕОРЕТИЧНІ ВІДОМОСТІ ПРО ДВОБІЧНУ ЧЕРГУ 8](#_Toc104925570)

[1.1 Двобічна черга 8](#_Toc104925571)

[1.2 Основні дії над чергою 8](#_Toc104925572)

[2 ПРОГРАМНА РЕАЛІЗАЦІЯ ДВОБІЧНОЇ ЧЕРГИ 10](#_Toc104925573)

[2.1 Структури двобічної черги та дискотеки 10](#_Toc104925574)

[2.2 Функції та їх призначення 10](#_Toc104925575)

[3 ІНСТРУКЦІЯ КОРИСТУВАЧА 12](#_Toc104925576)

[3.1 Меню 12](#_Toc104925577)

[3.2 Користувач ввів 1 13](#_Toc104925578)

[3.3 Користувач ввів 2 13](#_Toc104925579)

[3.4 Користувач ввів 3 14](#_Toc104925580)

[3.5 Користувач ввів 4 14](#_Toc104925581)

[Видалення 1-го елементу черги 14](#_Toc104925582)

[Видалення 1-го елементу 14](#_Toc104925583)

[3.6 Користувач ввів 5 14](#_Toc104925584)

[3.7 Користувач ввів 6 15](#_Toc104925585)

[3.8 Користквач ввів 7 15](#_Toc104925586)

[Пошук найтривалішого за звучанням альбому 15](#_Toc104925587)

[Пошук найтривалішого за звучанням альбому 15](#_Toc104925588)

[3.9 Користувач ввів 8 16](#_Toc104925589)

[3.10 Користувач ввів 9 18](#_Toc104925590)

[3.11 Користувач ввів 10 18](#_Toc104925591)

[3.12 Користувач ввів 11 19](#_Toc104925592)

[Виведення списку з роком випуску пізніше, ніж введений 19](#_Toc104925593)

[Виведення списку з роком випуску пізніше, ніж введений 19](#_Toc104925594)

[3.13 Користувач ввів 12 19](#_Toc104925595)

[3.14 Користувач ввів 13 20](#_Toc104925596)

[ВИСНОВКИ 21](#_Toc104925597)

[ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ 22](#_Toc104925598)

[ДОДАТОК А КОД ПРОГРАМИ 23](#_Toc104925599)

# ВСТУП

Метою курсової роботи є закріплення і поглиблення знань, одержаних студентами в курсі «Алгоритмізація та програмування», розвиток навичок при виборі представлення початкових даних, вдосконалення техніки використання засобів тестування і налагоджування програми, грамотне оформлення документації на програмну розробку. В сучасний час під час програмування досить часто доводиться працювати з структурами даних. Існують як динамічні структури даних, так і статичні. У статичні відразу після запису резервується і закріплюється за структурою виділена область пам'яті. У динамічних структурах пам’ять може як резервуватися, так і звільнятися під час роботи програми. Набагато зручніше працювати саме з динамічними структурами даних.

Існують такі динамічні структури даних:

* список (однозв’язний, двозв’язний);
* черга (проста, кільцева, з пріоритетами);
* стек;
* дерево.

# 1 ТЕОРЕТИЧНІ ВІДОМОСТІ ПРО ДВОБІЧНУ ЧЕРГУ

## 1.1 Двобічна черга

Двобічна черга - це особливий тип даних у галузі комп’ютерного програмування. У цьому абстрактному типі даних елементи можна додавати як з передньої, так і з задньої сторони черги. Завдяки цій властивості він також відомий як список, пов'язаний з головою та хвостом. У цьому черга насправді символізує сукупність елементів, розташованих аналогічно лінії з двома кінцями. Елементи можна додавати або видаляти з будь-якого кінця без будь-яких типів обмежень. Багато мов програмування використовують цю методику через її безліч застосувань.

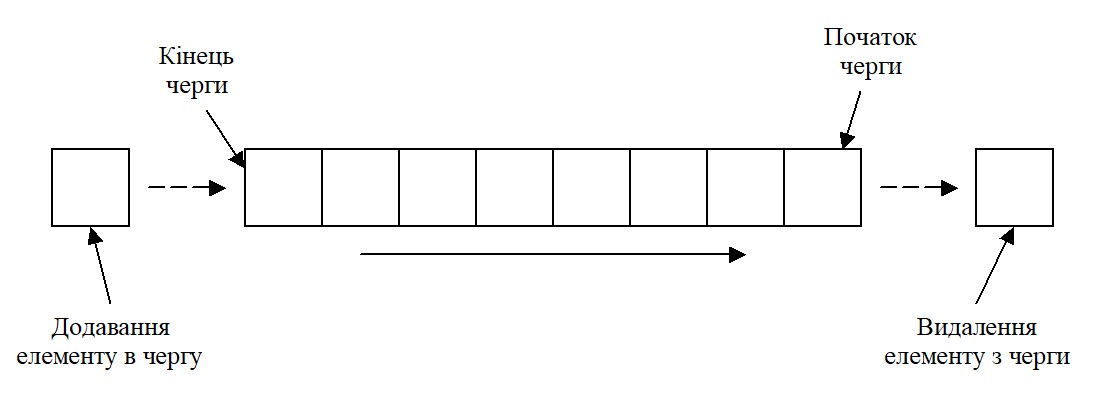
## 1.2 Основні дії над чергою

Типові операції:

* додавання елемента в кінець черги;
* додавання елемента в початок черги;
* вибірка останнього елемента;
* вибірка першого елемента;
* перевірка першого елемента (без видалення з деку);
* перевірка останнього елемента (без видалення з деку).

Розглянемо алгоритм додавання елементу до динамічної структури даних

двобічна черга(рис.1.1)



Двобічна черга - рис 1.1

На рисунку 1.1 досить зрозуміло показано, як працює додавання, але можу додати, видалення працює так само.

# 2 ПРОГРАМНА РЕАЛІЗАЦІЯ ДВОБІЧНОЇ ЧЕРГИ

## 2.1 Структури двобічної черги та дискотеки

1. Структура двобічної черги:

struct queue

{

struct disc \*first; //Показчик на 1-й елемент

struct disc \*last; //Показчик на останній елемент

};

2. Структура дискотеки:

struct disc //Структура данных

{

char name[35]; //Назва альбому

int year; //Рік випуску

float duration; //Тривалість

char author[35]; //Автор

struct disc \*next; //Показчик на наступний елемент

struct disc \*prev; //Показчик на попередній елемент

};

## 2.2 Функції та їх призначення

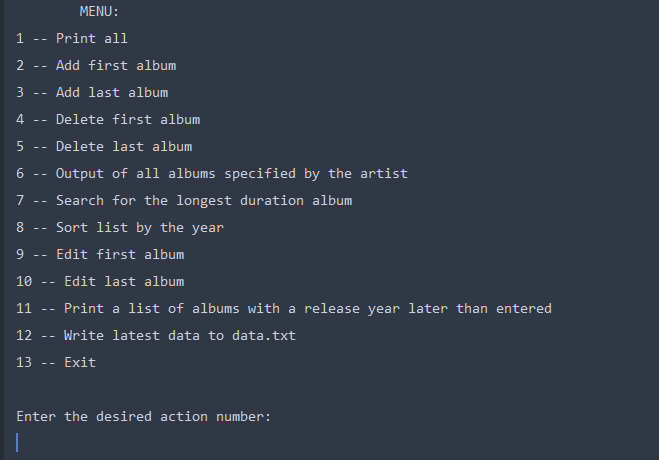
Функції, які були розроблені в ході виконня курсової роботи:

1. void Enter(struct disc \*a) – функція для вводу данних. Дана функція використовувалась майже в кожній із наведених. Та є однією з найважливіших.
2. void init\_queue(struct queue \*q) – функція ініціалізації черги. Без даної функції користувач не зможе взаємодіяти з данними.
3. void add\_first(struct queue \*q) – функція, яка додає елемент на початок черги. Вона являється однією з основних для черги.
4. void remove\_first(struct queue \*q) – функція, яка видаляє перший елемент черги.
5. void add\_last(struct queue \*q) – функція, яка додає елемент в кінець черги. Вона являється однією з основних для двобічної черги.
6. void remove\_last(struct queue \*q) – функція, яка видаляє останній елемент черги.
7. void print(struct queue \*q) - функія, яка виводить всі данні на екран. Була використана в багатьох з наведених функцій.
8. void free\_queue(struct queue \*q) – функція для очистки данних, після виходу із програми.
9. void search\_artist(struct queue \*q) – функція пошуку альбомів, обраного виконавця.
10. void search\_max(struct queue \*q) – функція пошуку найтривалішої композиції.
11. void sort\_words(char \*prev, char \*next) – допоміжна функція для (void sort\_by\_year(struct queue \*q)), була створенна для зменшення обьєму коду.
12. void sort\_by\_year(struct queue \*q) – функція сортування композицій за роком випуску.
13. void edit\_mas\_first(struct queue \*q) – функція коригування 1-го елементу черги.
14. void edit\_mas\_last(struct queue \*q) – функція киригування останнього елементу черги.
15. void print\_by\_year(struct queue \*q) – функція виводить композиції, які були випущені пізніше введеного року.
16. void write\_data(struct queue \*q) – функція запису кінцевих данних в файл типу (data.txt).

# 3 ІНСТРУКЦІЯ КОРИСТУВАЧА

## 3.1 Меню

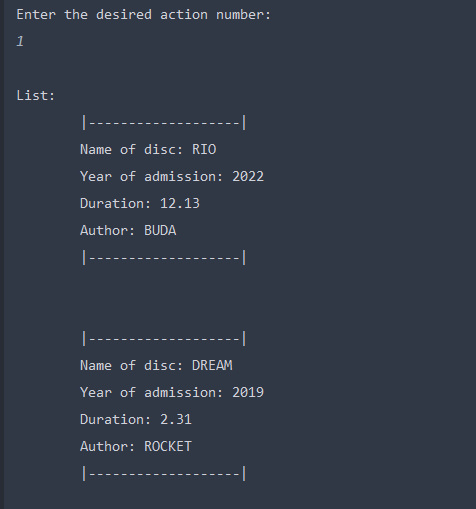
Програма при запуску одразу виводить меню та дає користувачу вибір 13 операцій, перегляд меню (рис 3.1). Якщо користувач введе число не відповідаюче меню, то програма буде надавати нові спроби.



Меню - рис 3.1

## 3.2 Користувач ввів 1

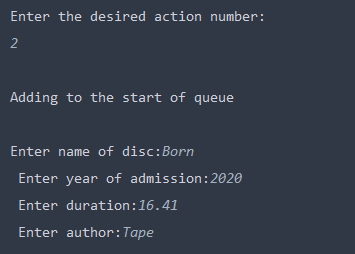
Вивести всі данні на екран можна побачити на рис 3.2



Виведення данних на екран - рис 3.2

## 3.3 Користувач ввів 2

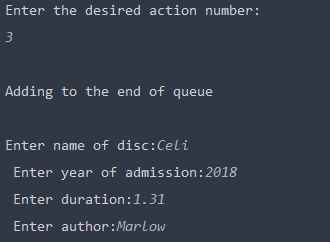
Додавання елементу на початок можна побачити на рис 3.3



Додавання елементу в початок - рис 3.3

## **3.4 Користувач ввів 3**

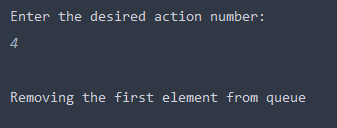
Додавання елементу в кінець черги можна побачити на рис 3.4



Додавання елементу в кінець - рис 3.4

## **3.5 Користувач ввів 4**

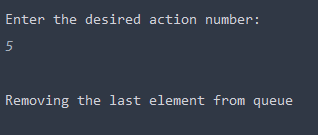
Видалення 1-го елементу черги можна побачити на рис 3.5



Видалення 1-го елементу - рис 3.5

## 3.6 Користувач ввів 5

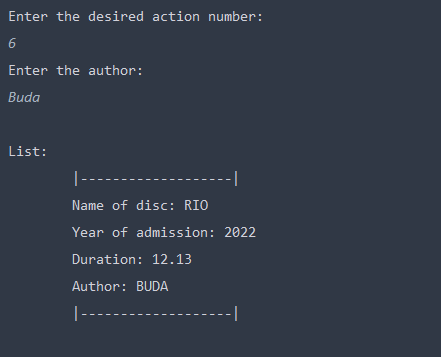
Видалення останньго елементу черги можна побачити на рис 3.6



Видалення останньго елементу - рис 3.6

## 3.7 Користувач ввів 6

Виведення всіх альбомів, заданого виконавця показано на рис 3.7

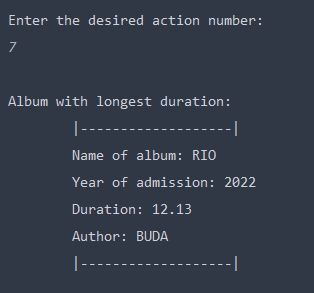


Виведення всіх альбомів, заданого виконавця - рис 3.7

# 

## **3.8 Користквач ввів 7**

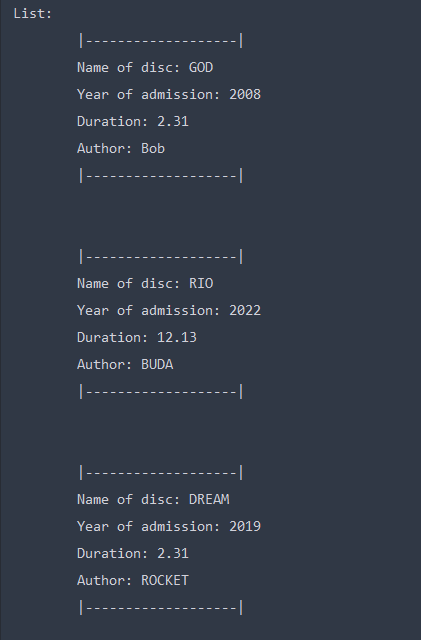
Пошук найтривалішого за звучанням альбомупоказано на рис 3.8



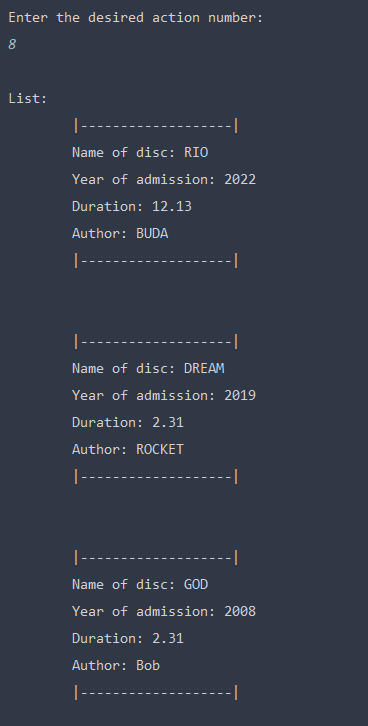
Пошук найтривалішого за звучанням альбому - рис 3.8

## 3.9 Користувач ввів 8

Сортування по полю «рік випуску» показано на рис 3.9 - 3.10



Данні до сортування - рис 3.9

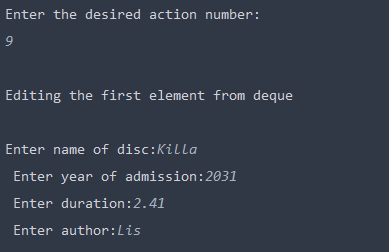


Данні після сортування - рис 3.10

# 

## 3.10 Користувач ввів 9

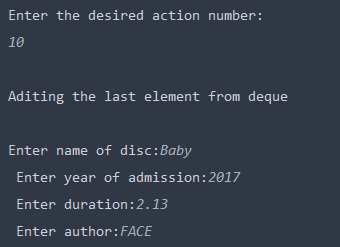
Коригування 1–го елементу показано на рис 3.11



Коригування 1–го елементу - рис 3.11

## 3.11 Користувач ввів 10

Коригування останнього елементу показано на рис 3.12

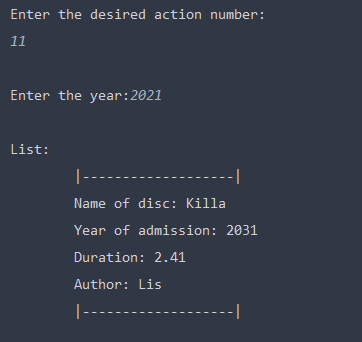


Коригування останнього елементу - рис 3.12

## **3.12 Користувач ввів 11**

Виведення списку з роком випуску пізніше, ніж введений показано на

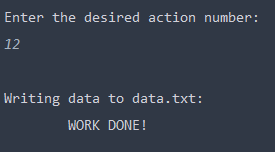
рис 3.13



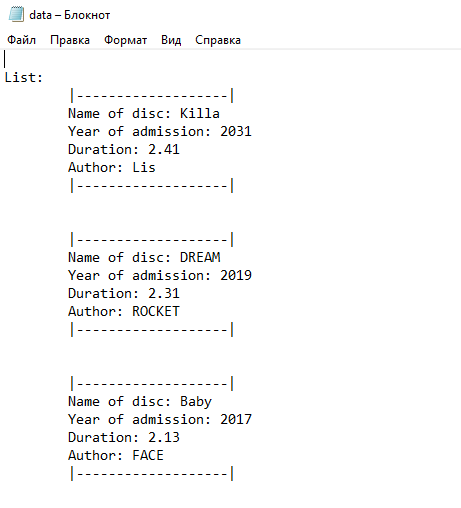
Виведення списку з роком випуску пізніше, ніж введений - рис 3.13

## 3.13 Користувач ввів 12

Запис данних в файл показано на рис 3.13 – 3.14



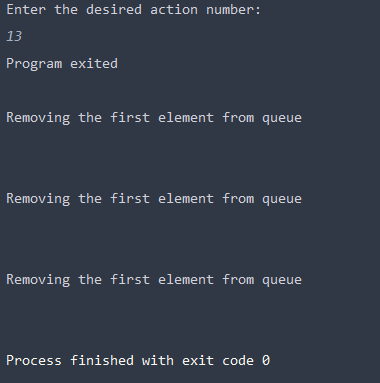
Процес запису - рис 3.13



Записані данні - рис 3.14

## 3.14 Користувач ввів 13

Вихід із програми показано на рис 3.15



Вихід із програми - рис 3.15

# ВИСНОВКИ

Було виконано завдання до курсової роботи в повному обсязі з дотриманням всіх вимог до виконання.

Поглиблено знання у роботі з динамічними структурами даних, програмними функціями, покажчиками, виділенням та видаленням зайнятої пам’яті.

Під час вивчення матеріалів для виконання курсової роботи були поглиблені теоретичні знання у програмуванні на мові С, закріплені на практиці навички у роботі зі структурами, функціями.

В ході роботи неодноразово виникали труднощі, пов’язані з неправильним написанням коду програми та нерозумінням таких моментів, як сортування та пошук найтривалішого за звучання альбому. Труднощі були вирішені завдяки інтернет-ресурсам.

Свій вклад у виконання курсової роботи оцінено в 75 %.

# ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Відомості про двобічну чергу. Двобічна черга: веб-сайт. URL: [https://uk.wikipedia.org/wiki/](https://uk.wikipedia.org/wiki/%D0%94%D0%B2%D0%BE%D0%B1%D1%96%D1%87%D0%BD%D0%B0_%D1%87%D0%B5%D1%80%D0%B3%D0%B0) (дата звернення: 20.04.2022).

2. Що таке двобічна черга: веб-сайт. URL:

<https://uk.theastrologypage.com/double-ended-queue>

(дата звернення: 18.04.2022)

3. Динамічні типи данних: веб-сайт. URL:

<http://mmsa.kpi.ua/sancho/ASD_HTM/DynaIntr.html> (дата звернення: 26.04.2022).

# ДОДАТОК А КОД ПРОГРАМИ