

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



Звіт

про виконання

Лабораторних та практичних робіт № 5

з дисципліни: «Мови та парадигми програмування»

з розділу: «Ерік 5. Файли. Системи числення. Бінарні файли. Символи та рядкові змінні. Текстові файли. Стандартна бібліотека. Створення й використання бібліотек»

Виконала:

студентка групи ШІ-14 - Гуменюк Тетяна В'ячеславівна

Тема роботи:

Вивчення файлів, бінарних та текстових файлів, систем числення. Ознайомлення з символами, рядковими змінними, стандартною бібліотекою.

Мета роботи:

Створення й використання бібліотек, файлів, застосування знань про символи та рядкові змінні.

Теоретичні відомості:

- 1) Теоретичні відомості з переліком важливих тем:
 - Тема №1: Файли
 - Тема №2: Системи числення
 - Тема №3: Символи та рядкові змінні
 - Тема №4: Бібліотека C++
- 2) Індивідуальний план опрацювання теорії:
 - Тема №1: Файли
 - Джерела Інформації:
 - <https://www.youtube.com/watch?v=SSNJ7alki-E>
 - <https://youtu.be/2lzVB8bkM8o?si=t42wYDcbb9DbI6Rg>
 - <https://www.youtube.com/watch?v=2lzVB8bkM8o>
 - Що опрацьовано:
 - Текстові та бінарні файли
 - Статус: Ознайомлена
 - Початок опрацювання теми: 19 грудня
 - Звершення опрацювання теми: 22 грудня
 - Тема №2: Системи числення
 - Джерела Інформації:
 - <https://youtu.be/c5FSYMPsLN0?si=yZgsgT9ktrN7x6PI>
 - Що опрацьовано:
 - Переведення чисел в різні системи числення
 - Статус: Ознайомлена
 - Початок опрацювання теми: 23 грудня
 - Звершення опрацювання теми: 23 грудня
 - Тема №3: Символи та рядкові змінні
 - Джерела Інформації:
 - <https://naurok.com.ua/prezentaciya-do-uroku-simvolni-ta-ryadkovi-velichini-123565.html>
 - http://www.kievoit.ippo.kubg.edu.ua/kievoit/2016/73_C++/index.html
 - Що опрацьовано:
 - Функції для роботи з символами та рядковими змінними
 - Статус: Ознайомлена
 - Початок опрацювання теми: 24 грудня
 - Звершення опрацювання теми: 26 грудня
 - Тема №4: Бібліотека
 - Джерела Інформації:
 - <https://acode.com.ua/urok-108-perevantazhennya-funktsij/>
 - <https://www.youtube.com/watch?v=m-WJikuZGuU>
 - <https://youtu.be/5mD-rhaYF4U?si=SC8LpIbG-jhEeNLI>
 - <https://www.youtube.com/watch?v=mnwDpO4zqLA>
 - Що опрацьовано:

- Стандартна бібліотека C++
- Створення статичної бібліотеки
- Статус: Ознайомлена
- Початок опрацювання теми: 26 грудня
- Звершення опрацювання теми: 2 грудня

Виконання роботи:

1. Опрацювання завдання та вимог до програм та середовища:

- Завдання №1: Algotester lab 4 task 2
- Деталі завдання:

Вам дано масив a з N цілих чисел.

Спочатку видаліть масиву a усі елементи що повторюються, наприклад масив $[1, 3, 3, 4]$ має перетворитися у $[1, 3, 4]$.

Після цього оберніть посортовану версію масиву a на K , тобто при $K = 3$ масив $[1, 2, 3, 4, 5, 6, 7]$ перетвориться на $[4, 5, 6, 7, 1, 2, 3]$.

Виведіть результат.

- Завдання №2: Algotester lab 6 task 2
- Деталі завдання:

У вас є шахова дошка розміром 8×8 та дуже багато фігур.

Кожна клітинка може мати таке значення:

- Пуста клітинка O
- Пішак P
- Тура R
- Кінь N
- Слон B
- Король K
- Королева Q

Вам дають позиції фігур на дошці (всі фігури одного кольору, кількість королів може бути > 1).

Далі йдуть Q запитів з координатами клітинки $\{x, y\}$. На кожен запит ви маєте вивести стрічку s_i - посортовані за алфавітом букви фігур, які атакують цю клітинку (пішаки атакують вниз).

У випадку, якщо на клітинці стоїть якась фігура - виведіть символ X .

У випадку, якщо клітинку не атакують - виведіть O .

Наявність фігури у певній клітинці не блокує атаку для іншої фігури. Тобто якщо між турою та клітинкою стоїть інша фігура - вважається що тура атакує цю клітинку.

- Важливі деталі для врахування в імплементації: якщо фігуру атакує дві однакових фігури то виводиться лише одна

- Завдання №3: Class practice

- Деталі завдання:

Реалізувати функцію створення файлу і запису в нього даних:

```
enum FileOpResult { Success, Failure, ... };
```

```
FileOpResult write_to_file(char *name, char *content);
```

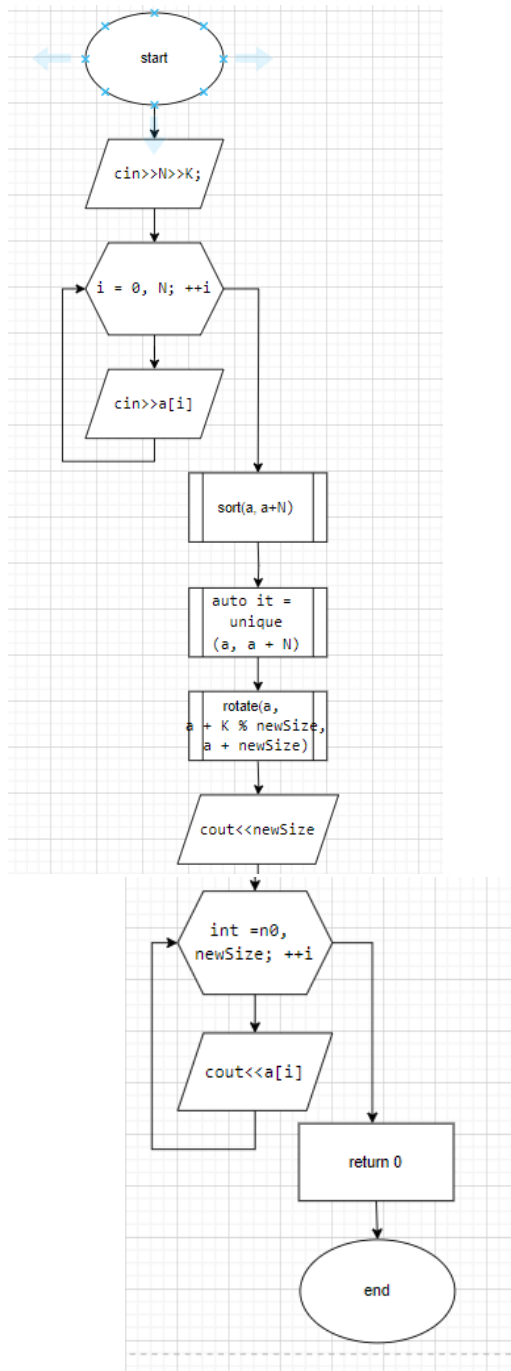
Умови задачі:

- створити файл із заданим ім'ям; якщо файл існує – перезаписати його вміст
- написати код стійкий до різних варіантів вхідних параметрів
- name – ім'я, може не включати шлях
- записати у файл вміст стрічки content, прочитати content із стандартного вводу
- повернути статус операції: Success – все пройшло успішно, Failure – файл не вдалося створити, або збій операції відкриття, запису даних, чи закриття файла.

2. Дизайн та планована оцінка часу виконання завдань:

Програма №1: Algotester lab 4 task 2

- Блок-схема



- Планований час на реалізацію – 3 години

3. Код програм:

Завдання №1: Algotester lab 4 task 2

```

#include <iostream>
#include <algorithm>
using namespace std;
int main() {
    int N, K;
    cin >> N >> K;
    int a[N];
    for (int i = 0; i < N; ++i) {
        cin >> a[i];
    }
    sort(a, a+N);
    auto it = unique(a, a + N);

```

```

    int newSize = distance(a, it);
    rotate(a, a + K % newSize, a + newSize);
    cout << newSize << '\n';
    for (int i = 0; i < newSize; ++i) {
        cout << a[i] << ' ';
    }
    return 0;
}

```

Завдання №2: Algotester lab 6 task 2

```

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

bool pawn (int xf, int yf, int x0, int y0)
{
    return abs(xf-x0) == 1 && yf+1 == y0;
}

bool tour(int xf, int yf, int x0, int y0)
{
    return xf==x0 || yf==y0;
}

bool bishop(int xf, int yf, int x0, int y0)
{
    return abs(x0-xf) == abs(y0-yf);
}

bool queen(int xf, int yf, int x0, int y0)
{
    return tour(xf, yf, x0, y0) || bishop(xf, yf, x0, y0);
}

bool horse(int xf, int yf, int x0, int y0)
{
    return (abs(x0-xf) == 2 && abs(y0-yf) == 1) || (abs(x0-xf) == 1 && abs(y0-yf) == 2);
}

bool king(int xf, int yf, int x0, int y0)
{
    //return (abs(x0-xf) == 1 && abs(y0-yf) == 1);
    return (abs(x0-xf) == 1 && abs(y0-yf) == 1 || abs(x0-xf) == 1 && abs(y0-yf) == 0 || abs(x0-xf) == 0
&& abs(y0-yf) == 1);
}

bool procesing_piece(char piece, int x, int y, int x0, int y0)
{
    switch(piece)
    {
        case 'K':
            return king(x,y,x0,y0);
        case 'N':
            return horse(x,y,x0,y0);
        case 'R':
            return tour(x,y,x0,y0);
        case 'P':
            return pawn(x,y,x0,y0);
        case 'B':
            return bishop(x,y,x0,y0);
        case 'Q':

```

```

        return queen(x,y,x0,y0);
    }
    return false;
}
void print_results(string results[], int results_count)
{
    for(int i=0; i<results_count; i++)
    {
        if(results[i] == "")
        {
            cout<<"0"<<endl;
        }
        else
        {
            sort(results[i].begin(), results[i].end());
            cout<<results[i]<<endl;
        }
    }
}
int main()
{
    string board[8];
    for(int row = 0; row <8; row++)
    {
        cin>>board[row];
    }
    int request_count;
    cin>> request_count;
    int requests[request_count][2];
    for(int i = 0; i<request_count; i++)
    {
        cin>>requests[i][1];
        cin>>requests[i][0];
    }
    string results[request_count];
    for(int row = 0; row <8; row++)
    {
        for(int column = 0; column<8; column++)
        {
            char piece = board[row][column];
            if(piece!= '0')
            {
                for(int i=0; i<request_count; i++)
                {
                    int x = requests[i][0]-1;
                    int y = requests[i][1]-1;
                    if(board[y][x] != '0')
                    {
                        results[i] = "X";
                    }
                    else if(procesing_piece(piece, column, row, x, y))
                    {
                        if (count(results[i].begin(), results[i].end(), piece) == 0) {
                            results[i] += piece;
                        }
                    }
                }
            }
        }
    }
    print_results(results, request_count);
}

```

Завдання №3: Class practice

```
#include <iostream>
#include <fstream>
using namespace std;
enum FileOpResult {Success, Failure};
FileOpResult write_to_file(const char *name, char *content)
{
    ofstream file_out(name);
    if(file_out.is_open())
    {
        file_out<<content;
        file_out.close();
        return Success;
    }
    return Failure;
}

int main()
{
    cout<<"enter file name"<<endl;
    char* file_name;
    cin>>file_name;
    cout<<"enter text"<<endl;
    char* text;
    cin>>text;
    FileOpResult result = write_to_file(file_name, text);
    switch (result)
    {
        case Success:
            cout<<"Success";
            break;
        case Failure:
            cout<<"Failure";
            break;
    }
}
```

4. Результати виконання завдань, тестування та фактично затрачених час:

Завдання №1: Algotester lab 4 task 2

```
4 6
3 1 4 5
4
4 5 1 3
```

Час затрачених на виконання завдання: 2 години

Завдання №2: Algotester lab 6 task 2

```
000R0000
OK000000
OR000000
00000000
0000B000
00000000
Q0000000
00000000
4
1 1
1 4
3 4
5 3
X
R
R
Q
```

Час затрачений на виконання завдання: 5 годин

Завдання №3: Class practice

```
enter file name
hello
```

Час затрачений на виконання завдання: 3 години