

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



Звіт

про виконання лабораторних та практичних робіт блоку № 5

На тему: «Файли. Бінарні Файли. Символи і Рядкові Змінні та Текстові Файли.
Стандартна бібліотека та деталі/методи роботи з файлами. Створення й
використання бібліотек.»

з дисципліни: «Основи програмування»

до:

ВНС Лабораторної Роботи № 6
ВНС Лабораторної Роботи № 8
ВНС Лабораторної Роботи № 9
Алготестер Лабораторної Роботи №4
Алготестер Лабораторної Роботи №6
Практичних Робіт до блоку №5

Виконав:

Студент групи ШІ-11
Климчук Юрій Олегович

Львів 2024

Тема роботи: Файли. Бінарні Файли. Символи і Рядкові Змінні та Текстові Файли. Стандартна бібліотека та деталі/методи роботи з файлами. Створення й використання бібліотек

Мета роботи: Ознайомитися з принципами роботи з файлами в програмуванні, зокрема бінарними та текстовими файлами. Навчитися використовувати символи, рядкові змінні та текстові файли у програмах. Розглянути функціонал стандартної бібліотеки для роботи з файлами, а також методи створення та використання власних бібліотек для роботи з файлами.

Теоретичні відомості:

1)Перелік тем:

1. Вступ до Роботи з Файлами;
2. Символи і Рядкові Змінні;
3. Текстові Файли;
4. Бінарні Файли;
5. Стандартна бібліотека та робота з файлами;
6. Створення й використання бібліотек.

2)Індивідуальний план опрацювання теорії:

1. Вступ до Роботи з Файлами

<https://acode.com.ua/urok-220-bazovyj-fajlovyj-vvid-i-vyvvid/>

2. Символи і Рядкові Змінні

https://www.w3schools.com/cpp/cpp_files.asp

3. Текстові Файли

<https://www.geeksforgeeks.org/file-handling-c-classes/>

4. Бінарні Файли

<https://www.geeksforgeeks.org/difference-between-cpp-text-file-and-binary-file/>

5. Стандартна бібліотека та робота з файлами

- Unordered map: [unordered_map in C++ STL - GeeksforGeeks](https://www.geeksforgeeks.org/unordered-map-in-cpp-stl/)
- Стандарти бібліотеки і їх виклик:
https://en.cppreference.com/w/cpp/standard_library

6. Створення й використання бібліотек

<https://stackoverflow.com/questions/16693273/how-do-i-create-a-library>

<https://www.geeksforgeeks.org/how-do-i-create-a-library-in-cpp/>

Виконання роботи:

1)Перелік завдань:

- John Black - Epic 5 Task 1 - Theory Education Activities
- John Black - Epic 5 Task 2 - Requirements management (understand tasks) and design activities (draw flow diagrams and estimate tasks 3-9)
- John Black - Epic 5 Task 3 - Lab# programming: VNS Lab 6(варіант 4)
- John Black - Epic 5 Task 4 - Lab# programming: VNS Lab 8(варіант 4)
- John Black - Epic 5 Task 5 - Lab# programming: VNS Lab 9(варіант 4)
- John Black - Epic 5 Task 6 - Lab# programming: Algotester Lab 4(варіант 1)
- John Black - Epic 5 Task 7 - Lab# programming: Algotester Lab 6(варіант 1)
- John Black - Epic 5 Task 8 - Practice# programming: Class Practice Task
- John Black - Epic 5 Task 9 - Practice# programming: Self Practice Task
- John Black - Epic 5 Task 10 - Result Documentation Report and Outcomes Placement Activities (Docs and Programs on GitHub)
- John Black - Epic 5 Task 11 - Results Evaluation and Release

2)Умови завдань:

Task 3: Задано рядок, що складається із символів. Символи поєднуються в слова. Слова одне від одного відокремлюються одним або декількома пробілами. Наприкінці тексту ставиться крапка. Текст містить не більше 255 символів. Виконати ввід рядка, використовуючи функцію gets(s) і здійснити обробку рядка у відповідності зі своїм варіантом.

Надрукувати всі слова, які співпадають з її першим словом.

Task 4: Сформувати двійковий файл із елементів, заданої у варіанті структури, роздрукувати його вміст, виконати знищення й додавання елементів у відповідності зі своїм варіантом, використовуючи для пошуку елементів що знищуються чи додаються, функцію. Формування, друк, додавання й знищення елементів оформити у вигляді функцій. Передбачити повідомлення про помилки при відкритті файлу й виконанні операцій вводу/виводу.

Структура "Людина":

- прізвище, ім'я, по батькові;
- домашня адреса;
- номер телефону;
- вік.

Знищити усі елементи із заданим віком, додати елемент після елемента із заданим номером.

Task 5: Створити текстовий файл F1 не менше, ніж з 10 рядків і записати в нього інформацію. Виконати завдання.

- 1) Скопіювати з файлу F1 у файл F2 рядки, починаючи з 4.
- 2) Підрахувати кількість символів в останньому слові F2.

Task 6: Lab 4v1

Обмеження: 1 сек., 256 MiB

Вам дано 2 цілих чисел масиви, розміром N та M.
Ваше завдання вивести:

1. Різницю N-M
2. Різницю M-N
3. Їх перетин
4. Їх об'єднання
5. Їх симетричну різницю

Вхідні дані

У першому рядку ціле число N - розмір масиву 1
У другому рядку N цілих чисел - елементи масиву 1
У третьому рядку ціле число M - розмір масиву 2
У четвертому рядку M цілих чисел - елементи масиву 2

Вихідні дані

Вивести результат виконання 5 вищезазначених операцій у форматі:

У першому рядку ціле число N - розмір множини
У наступному рядку N цілих чисел - посортована у порядку зростання множина
Пам'ятайте, ви маєте написати 2 варіанти розв'язку, один з використанням засобів STL (std::set_intersection, std::set_symmetric_difference, std::set_difference, std::set_union), інший зі своєю реалізацією. Своє сортування можна не писати.

Task 7: Lab 6v1

Обмеження: 2 сек., 256 MiB

Вам дано N слів та число K.

Ваше завдання перерахувати букви в словах, які зустрічаються в тексті більше-рівне ніж K разів (саме слово, не буква!).

Великі та маленькі букви вважаються однаковими, виводити необхідно малі, посортовані від останньої до першої у алфавіті. Букву потрібно виводити лише один раз.

У випадку якщо таких букв немає - вивести "Empty!".

Вхідні дані

Цілі числа N та K - загальна кількість слів та мінімальна кількість слів щоб враховувати букви цього слова в результаті.

N стрічок s

Вихідні дані

У першому рядку ціле число M - кількість унікальних букв

У другому рядку унікальні букви через пробіли

Task 8: Задача №1 – Запис текстової стрічки у файл із заданим ім'ям

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.

Задача №2 – Копіювання вмісту файла у інший файл

Реалізувати функцію створення файла і запису в нього даних:

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

```
FileOpResult copy_file(char *file_from, char *file_to);
```

Умови задачі:

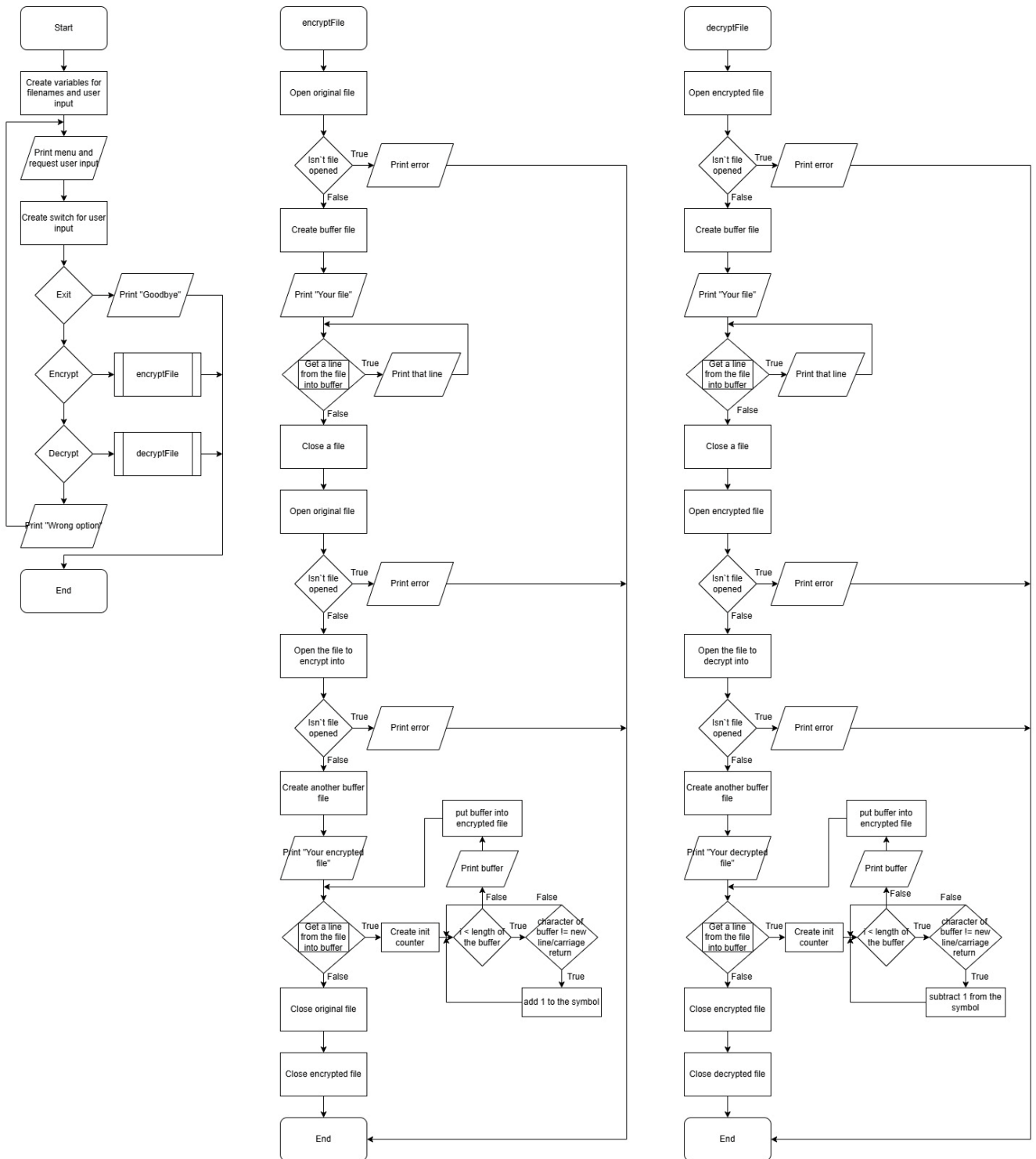
- копіювати вміст файла з ім'ям file_from у файл з ім'ям file_to; написати код стійкий до різних варіантів вхідних параметрів, обробити всі можливі варіанти відсутності одного з файлів
- file_from, file_to – можуть бути повним або відносним шляхом
- повернути статус операції: Success – все пройшло успішно, Failure – файл не вдалося створити, або збій операції відкриття, читання чи запису даних, закриття файла.

Task 9: Шифрування/дешифрування файлу за рахунок зміщення символів

на одну позицію по ASCII таблиці.

3)Дизайн та планова оцінка часу виконання завдань:

Epic 5 Task 9 - Practice# programming: Self Practice Task



Орієнтовний час виконання: 55хв

4)Код програм з посиланням на зовнішні ресурси:

John Black - Epic 5 Task 3 - Lab# programming: VNS Lab 6(варіант 4)

Посилання на файл програми: https://github.com/artificial-intelligence-department/ai_programming_playground_2024/pull/397/files#diff-baefb119101c4bb413df85059eeb4584871bc4ffe7ed95f76e5d835497158c90

```
#include <stdio.h>
#include <string.h>

void Coincidences(char str[]){
    char *token;
    token = strtok(str, " ");
    char *first_word = token;
    printf("The first word: %s\n", first_word);
    while(token != NULL){
        token = strtok(NULL, " ");
        if(token != NULL && strcmp(first_word, token) == 0){
            printf("%s\n", token);
        }
    }
}

int main() {
    char s[256];
    char space = ' ';
    fgets(s, sizeof(s), stdin);
    s[strlen(s) - 1] = ' ';
    Coincidences(s);
    return 0;
}
```

John Black - Epic 5 Task 4 - Lab# programming: VNS Lab 8(варіант 4)

Посилання на файл програми: https://github.com/artificial-intelligence-department/ai_programming_playground_2024/pull/397/files#diff-fb1337cd7917a6a4e108a439e8375f27f11db2001001e2cb733e9285e86e6a1f

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

#define FILE_NAME "people.dat"
```



```
//Структура для запису інформації про людину
typedef struct {
    char surname[50];
    char name[50];
    char patronymic[50];
    char address[100];
    char phone[15];
    int age;
} Person;

// Функція для додавання людини у файл
void addPerson() {
    FILE *file = fopen(FILE_NAME, "ab");
    if (!file) {
        perror("Error opening file");
        exit(1);
    }

    Person person;
    printf("Enter details of the person:\n");
    printf("Surname: "); scanf("%s", person.surname);
    printf("Name: "); scanf("%s", person.name);
    printf("Patronymic: "); scanf("%s", person.patronymic);
    printf("Address: "); scanf("%s", person.address);
    printf("Phone: "); scanf("%s", person.phone);
    printf("Age: "); scanf("%d", &person.age);

    fwrite(&person, sizeof(Person), 1, file);

    fclose(file);
    printf("Person added successfully.\n");
}

//Функція для виводу вмісту файлу
void printFile() {
    FILE *file = fopen(FILE_NAME, "rb");
    if (!file) {
        perror("Error opening file");
        exit(1);
    }
}
```

```

    Person person;
    printf("File contents:\n");
    while (fread(&person, sizeof(Person), 1, file)) {
        printf("Surname: %s, Name: %s, Patronymic: %s\n",
person.surname, person.name, person.patronymic);
        printf("Address: %s, Phone: %s, Age: %d\n", person.address,
person.phone, person.age);
        printf("-----\n");
    }

    fclose(file);
}

//Функція для видалення людини за віком
void deleteByAge(int age) {
    FILE *file = fopen(FILE_NAME, "rb");
    if (!file) {
        perror("Error opening file");
        exit(1);
    }
    //Створюємо тимчасовий файл
    FILE *temp = fopen("temp.dat", "wb");
    if (!temp) {
        perror("Error creating temporary file");
        fclose(file);
        exit(1);
    }

    //Записуємо у тимчасовий файл всіх людей окрім вказаного віку
    Person person;
    int found = 0;
    while (fread(&person, sizeof(Person), 1, file)) {
        if (person.age != age) {
            fwrite(&person, sizeof(Person), 1, temp);
        } else {
            found = 1;
        }
    }

    fclose(file);
    fclose(temp);
}

```

```

//замінюємо наш початковий тимчасовим файлом
remove(FILE_NAME);
rename("temp.dat", FILE_NAME);

if (found) {
    printf("Records with age %d have been deleted.\n", age);
} else {
    printf("No records found with age %d.\n", age);
}
}

//Функція для додавання людини після конкретного номера телефону
void addAfterPhone(char *phone) {
    FILE *file = fopen(FILE_NAME, "rb");
    if (!file) {
        perror("Error opening file");
        exit(1);
    }
    //Створюємо тимчасовий файл
    FILE *temp = fopen("temp.dat", "wb");
    if (!temp) {
        perror("Error creating temporary file");
        fclose(file);
        exit(1);
    }

    Person person, newPerson;
    int found = 0;

    printf("Enter details of the new person:\n");
    printf("Surname: "); scanf("%s", newPerson.surname);
    printf("Name: "); scanf("%s", newPerson.name);
    printf("Patronymic: "); scanf("%s", newPerson.patronymic);
    printf("Address: "); scanf("%s", newPerson.address);
    printf("Phone: "); scanf("%s", newPerson.phone);
    printf("Age: "); scanf("%d", &newPerson.age);

    //додаємо введену інформацію після запису із вказаним номером
    телефону
    while (fread(&person, sizeof(Person), 1, file)) {
        fwrite(&person, sizeof(Person), 1, temp);
        if (strcmp(person.phone, phone) == 0) {

```

```

        fwrite(&newPerson, sizeof(Person), 1, temp);
        found = 1;
    }
}

fclose(file);
fclose(temp);

//замінюємо наш початковий тимчасовим файлом
remove(FILE_NAME);
rename("temp.dat", FILE_NAME);

if (found) {
    printf("New record added after the record with phone number
%s.\n", phone);
} else {
    printf("No record found with phone number %s.\n", phone);
}
}

int main() {
    int choice, age;
    char phone[15];
    //реалізація меню
    while (1) {
        printf("\nMenu:\n");
        printf("1. Add a person\n");
        printf("2. Print file contents\n");
        printf("3. Delete records by age\n");
        printf("4. Add a record after a specific phone number\n");
        printf("5. Exit\n");
        printf("Choose an option: ");
        scanf("%d", &choice);

        switch (choice) {
            case 1:
                addPerson();
                break;
            case 2:
                printFile();
                break;
            case 3:

```

```

        printf("Enter age to delete records: ");
        scanf("%d", &age);
        deleteByAge(age);
        break;
    case 4:
        printf("Enter phone number: ");
        scanf("%s", phone);
        addAfterPhone(phone);
        break;
    case 5:
        exit(0);
    default:
        printf("Invalid choice.\n");
    }
}

return 0;
}

```

John Black - Epic 5 Task 5 - Lab# programming: VNS Lab 9(вариант 4)

Посилання на файл програми: https://github.com/artificial-intelligence-department/ai_programming_playground_2024/pull/397/files#diff-99c36b528a32a29e56674ac500e7a9366e3681abfd7667b102954c33e807e146

```

#include <iostream>
#include <cstring>

using namespace std;

//створюємо перший файл
void CreateFirstFile(const char* filename_1, const char** content){
    FILE *file_1 = nullptr;
    file_1 = fopen(filename_1, "w");
    if (file_1 == nullptr){
        cerr << "Failed to open a file";
        return;
    }

    for (int i = 0; i < 10; i++){
        fputs(content[i], file_1);
    }
}

```

```

    }
    fclose(file_1);
}
//записуємо у другий файл стрічки з 1 починаючи з 4-тої
void CreateSecondFile(const char* filename_1, const char*
filename_2){
    FILE *file_1 = nullptr;
    file_1 = fopen(filename_1, "r");
    if (file_1 == nullptr){
        cerr << "Failed to open the first file\n";
        return;
    }

    FILE *file_2 = nullptr;
    file_2 = fopen(filename_2, "w");
    if (file_2 == nullptr){
        cerr << "Failed to open the second file\n";
        fclose(file_1);
        return;
    }

    char f1_line[256];
    int line_index = 1;
    while(fgets(f1_line, sizeof(f1_line), file_1) != nullptr)
    {
        if(line_index >= 4)
        {
            fputs(f1_line, file_2);
        }
        line_index++;
    }

    fclose(file_1);
    fclose(file_2);
}
//рахуємо символи в останньому слові 2 файлу
void CountSymbols(const char* filename_2){
    FILE *file_2 = nullptr;
    file_2 = fopen(filename_2, "r");
    if (file_2 == nullptr){
        cerr << "Failed to open the second file\n";

```

```

        return;
    }

    char f2_line[256];
    char* last_word;
    int index = 1;

    while(fgets(f2_line, sizeof(f2_line), file_2) != nullptr);
    char* lastSpace = strrchr(f2_line, ' ');
    last_word = lastSpace + 1;
    cout << "The number of symbols in last word of the second file is
" << strlen(last_word);
}
int main(){
    //створюємо змінні для назв файлів та тексту
    const char* filename_1 = "vns_lab9_file1.txt";
    const char* filename_2 = "vns_lab9_file2.txt";
    const char* file_1_content[10] = {
        "I would say I'm sorry\n",
        "If I thought that it would change your mind\n",
        "But I know that this time\n",
        "I have said too much\n",
        "Been too unkind\n",
        "I tried to laugh about it\n",
        "Cover it all up with lies\n",
        "I tried to laugh about it\n",
        "Hiding the tears in my eyes\n",
        "Cause boys don't cry"
    };
    //по черзі викликаємо необхідні функції
    CreateFirstFile(filename_1, file_1_content);
    CreateSecondFile(filename_1, filename_2);
    CountSymbols(filename_2);
    return 0;
}

```

John Black - Epic 5 Task 6 - Lab# programming: Algotester Lab 4(варіант 1 з використанням STL)

Посилання на файл програми: https://github.com/artificial-intelligence-department/ai_programming_playground_2024/pull/397/files#diff-7fc9705b62f8d63fcee66cac6c5d4bbe5d88449ad09fce010c96fe3cdc00ffa

Посилання на алготестер: <https://algotester.com/uk/ContestProblem/DisplayWithEditor/135598>

```
#include <iostream>
#include <set>
#include <vector>
#include <algorithm>
using namespace std;

int main() {
    int N, M;

    //считуємо розмір та значення першої множини
    cin >> N;
    vector<int> n(N);
    for (int i = 0; i < N; i++) {
        cin >> n[i];
    }
    //считуємо розмір та значення другої множини
    cin >> M;
    vector<int> m(M);
    for (int i = 0; i < M; i++) {
        cin >> m[i];
    }

    //сортуюмо вектори
    sort(n.begin(), n.end());
    sort(m.begin(), m.end());

    //реалізуємо n - m різницю
    vector<int> difference;
    set_difference(n.begin(), n.end(), m.begin(), m.end(),
back_inserter(difference));
    cout << difference.size() << endl;
    for (size_t i = 0; i < difference.size(); i++) {
        if (i > 0) cout << " ";
        cout << difference[i];
    }
    cout << endl;
```



```

//реалізуємо m - n різницю
vector<int> reverse_difference;
set_difference(m.begin(), m.end(), n.begin(), n.end(),
back_inserter(reverse_difference));
cout << reverse_difference.size() << endl;
for (size_t i = 0; i < reverse_difference.size(); i++) {
    if (i > 0) cout << " ";
    cout << reverse_difference[i];
}
cout << endl;

//реалізуємо перетин
vector<int> intersection;
set_intersection(n.begin(), n.end(), m.begin(), m.end(),
back_inserter(intersection));
cout << intersection.size() << endl;
for (size_t i = 0; i < intersection.size(); i++) {
    if (i > 0) cout << " ";
    cout << intersection[i];
}
cout << endl;

//реалізуємо об'єднання
vector<int> union_set;
set_union(n.begin(), n.end(), m.begin(), m.end(),
back_inserter(union_set));
cout << union_set.size() << endl;
for (size_t i = 0; i < union_set.size(); i++) {
    if (i > 0) cout << " ";
    cout << union_set[i];
}
cout << endl;

//реалізуємо симетричну різницю
vector<int> symmetric_difference;
set_symmetric_difference(n.begin(), n.end(), m.begin(), m.end(),
back_inserter(symmetric_difference));
cout << symmetric_difference.size() << endl;
for (size_t i = 0; i < symmetric_difference.size(); i++) {
    if (i > 0) cout << " ";
    cout << symmetric_difference[i];
}

```

```
}  
cout << endl;  
  
return 0;  
}
```

John Black - Epic 5 Task 6 - Lab# programming: Algotester Lab 4(варіант 1 з власною імплементацією)

Посилання на файл програми: https://github.com/artificial-intelligence-department/ai_programming_playground_2024/pull/397/files#diff-13355ced1f96b4689870d76f1afff7129e8afff74d9ff04fd61d134606e3967d

Посилання на алготестер: <https://algotester.com/uk/ContestProblem/DisplayWithEditor/135598>

```
#include <iostream>  
#include <vector>  
#include <algorithm>  
  
using namespace std;  
  
//функція для виводу результатів операцій  
void PrintResult (const vector<int>& result){  
    cout << result.size() << endl;  
    for (int x : result) cout << x << " ";  
    cout << endl;  
}  
  
//функція для об'єднання множин  
vector<int> SetUnion(const vector<int>& a, const vector<int>& b) {  
    vector<int> result;  
    int i = 0, j = 0;  
    while (i < a.size() && j < b.size()) {  
        if (a[i] < b[j]) result.push_back(a[i++]);  
        else if (b[j] < a[i]) result.push_back(b[j++]);  
        else {  
            result.push_back(a[i]);  
            i++; j++;  
        }  
    }  
}
```

```

    }
    while (i < a.size()) result.push_back(a[i++]);
    while (j < b.size()) result.push_back(b[j++]);
    return result;
}

//функція для перетину множин
vector<int> Intersection(const vector<int>& a, const vector<int>& b)
{
    vector<int> result;
    int i = 0, j = 0;
    while (i < a.size() && j < b.size()) {
        if (a[i] < b[j]) i++;
        else if (b[j] < a[i]) j++;
        else {
            result.push_back(a[i]);
            i++; j++;
        }
    }
    return result;
}

//функція для знаходження різниці
vector<int> Difference(const vector<int>& a, const vector<int>& b){
    vector<int> result;
    int i = 0, j = 0;
    while (i < a.size() && j < b.size()) {
        if (a[i] < b[j]) result.push_back(a[i++]);
        else if (b[j] < a[i]) j++;
        else {
            i++;
            j++;
        }
    }
    while (i < a.size()) result.push_back(a[i++]);
    return result;
}

//функція для симетричної різниці (з використанням попередніх
функцій)
vector<int> SymmetricDifference(const vector<int>& a, const
vector<int>& b) {

```

```

    vector<int> result1 = Difference(a, b);
    vector<int> result2 = Difference(b, a);
    return SetUnion(result1, result2);
}

int main() {
    int N, M;

    //считуємо розмір та значення першої множини
    cin >> N;
    vector<int> n(N);
    for (int i = 0; i < N; i++) {
        cin >> n[i];
    }

    //считуємо розмір та значення другої множини
    cin >> M;
    vector<int> m(M);
    for (int i = 0; i < M; i++) {
        cin >> m[i];
    }

    //сортуємо вектори
    sort(n.begin(), n.end());
    sort(m.begin(), m.end());

    //записуємо результати операцій
    vector<int> diff1 = Difference(n, m);
    vector<int> diff2 = Difference(m, n);
    vector<int> intersection = Intersection(n, m);
    vector<int> union_set = SetUnion(n, m);
    vector<int> sym_diff = SymmetricDifference(n, m);

    //виводимо результати
    PrintResult(diff1);
    PrintResult(diff2);
    PrintResult(intersection);
    PrintResult(union_set);
    PrintResult(sym_diff);

    return 0;
}

```

John Black - Epic 5 Task 7 - Lab# programming: Algotester Lab 6(варіант 1)

Посилання на файл програми: https://github.com/artificial-intelligence-department/ai_programming_playground_2024/pull/397/files#diff-126172181eb674831953cab160463ef552278792721f06a7b6e2a843b99e3b6e

Посилання на алготестер:
<https://algotester.com/uk/ContestProblem/DisplayWithEditor/135604>

```
#include <iostream>
#include <string>
#include <unordered_map>
#include <set>
#include <vector>
#include <algorithm>
using namespace std;

//конвертуємо усі букви в нижній регістр
void LowerCase(string& str) {
    for (char& c : str) {
        c = tolower(c);
    }
}

int main() {
    //приймаємо значення для кількості слів
    int N, K;
    cin >> N >> K;

    //створюємо непосортовану хеш-таблицю для запису слів і їх
    //кількості
    unordered_map<string, int> wordCount;
    vector<string> words(N);
    //записуємо слова у вектор, паралельно переводимо у нижній
    //регістр і записуємо в хеш-таблицю
    for (int i = 0; i < N; i++) {
        cin >> words[i];
```

```

        LowerCase(words[i]);
        wordCount[words[i]]++;
    }

    //створюємо набір і записуємо туди слова які проходять
    //по кількості повторень
    set<char> validLetters;

    for (const auto& [word, count] : wordCount) {
        if (count >= K) {
            for (char c : word) {
                validLetters.insert(c);
            }
        }
    }

    //у разі відсутності таких слів виводимо "Пусто!"
    if (validLetters.empty()) {
        cout << "Empty!" << endl;
        return 0;
    }

    //записуємо слова з набору у вектор і сортуємо їх, у зворотньому
    //алфавітному порядку
    vector<char> sortedLetters(validLetters.begin(),
validLetters.end());
    sort(sortedLetters.rbegin(), sortedLetters.rend());

    //виводимо кількість таких літер і їх самих через пробіл
    cout << sortedLetters.size() << endl;
    for (size_t i = 0; i < sortedLetters.size(); i++) {
        cout << sortedLetters[i];
        if (i != sortedLetters.size() - 1) {
            cout << " ";
        }
    }
    cout << endl;

    return 0;
}

```

John Black - Epic 5 Task 8 - Practice# programming: Class Practice Task(завдання 1)

Посилання на файл програми: https://github.com/artificial-intelligence-department/ai_programming_playground_2024/pull/397/files#diff-a21eb9fala7ab5eaec5310a50345c45a79c8b4589ed020d8d1b7a7585ec7e0

```
#include <iostream>

using namespace std;

//створюємо перерахунок для результату нашої операції
enum FileOpResult { Success, Failure};

//функція типу нашого перерахунку для запису тексту у файл
FileOpResult write_to_file(char *name, char *content){
    //перевірка наявності ім'я файлу та тексту
    if (name == NULL || content == NULL){
        cout << "Error, no filename or content provided.\n";
        return Failure;
    }
    //перевірка чи файл відкрився
    FILE *file = fopen(name, "w");
    if (!file) {
        cout << "Error opening file\n";
        return Failure;
    }
    //перевірка чи записався текст у файл
    fwrite(content, sizeof(char), 1, file);
    if (file == NULL){
        cout << "Error while writing content into file.\n";
        return Failure;
    }

    //перевірка чи записався текст у файл
    if (fputs(content, file) == EOF){
        cout << "Error while writing content into file.\n";
        fclose(file);
        return Failure;
    }

    //перевірка чи закрився файл
    if(fclose(file) != 0){
```

```

        cout << "Error while closing the file \n";
        return Failure;
    }
    return Success;
}

int main() {
    //змінні для назви файлу та вмісту
    char filename[255];
    char file_content[1000];

    cout << "Enter the file name: ";
    cin >> filename;

    cout << "Enter the content of the file: ";
    cin >> file_content;

    //виклик функції і запис результату
    enum FileOpResult result = write_to_file(filename, file_content);

    //опрацювання результату
    if(result == 0){
        cout << "The operation is successful";
    }
    else{
        cout << "The operation failed";
    }
    return 0;
}

```

John Black - Epic 5 Task 8 - Practice# programming: Class Practice Task(завдання 2)

Посилання на файл програми: https://github.com/artificial-intelligence-department/ai_programming_playground_2024/pull/397/files#diff-7821ecfe265f3ea68a2197195763aba0deb0a0de021d9cd614ecdc0a64c8a68a

```

#include <iostream>

using namespace std;

//створюємо перерахунок для результату нашої операції
enum FileOpResult { Success, Failure};

```



```

//функція для копіювання вмісту одного файлу в інший
FileOpResult copy_file(char *file_from, char *file_to){
    //перевірка наявності імен файлів
    if (file_from == nullptr || file_to == nullptr){
        cout << "Error, no filenames provided";
        return Failure;
    }

    //перевірка чи файли відкрилися
    FILE *file_1 = fopen(file_from, "r");
    if (!file_1) {
        cout << "Error while opening the file " << file_from << endl;
        return Failure;
    }
    FILE *file_2 = fopen(file_to, "w");
    if (!file_2) {
        cout << "Error while opening the file " << file_to << endl;
        return Failure;
    }

    //порядковий запис файлу 1 у файл 2 з перевіркою на помилку
    char f1_line[256];
    while(fgets(f1_line, sizeof(f1_line), file_1) != nullptr)
    {
        if(fputs(f1_line, file_2) == EOF){
            cout << "Error while writing to file " << file_to <<
endl;
            fclose(file_1);
            fclose(file_2);
            return Failure;
        }
    }

    //перевірка чи закрилися файли
    if(fclose(file_1) != 0){
        cout << "Error while closing the file " << file_from << endl;
        return Failure;
    }
    if(fclose(file_2) != 0){
        cout << "Error while closing the file " << file_to << endl;
        return Failure;
    }
}

```

```

    }
    return Success;
}

int main() {
    //створюємо змінні з двома іменами файлів
    char file_from[64] = "file_from.txt";
    char file_to[64] = "file_to.txt";

    //виклик функції і запис результату
    enum FileOpResult result = copy_file(file_from, file_to);

    //опрацювання результату
    if(result == 0){
        cout << "The operation is successful";
    }
    else{
        cout << "The operation failed";
    }
    return 0;
}

```

John Black - Epic 5 Task 9 - Practice# programming: Self Practice Task

Посилання на файл програми: https://github.com/artificial-intelligence-department/ai_programming_playground_2024/pull/397/files#diff-8fc37e75b05fbe83da0b73c528469dc4e3dd93aa8dcf9f10dc1bdd61f698c18c

```

//Шифрування/дешифрування файлу за рахунок зміщення символів
//на одну позицію по ASCII таблиці

#include <cstdio>
#include <iostream>
#include <cstring>

using namespace std;

//файл для запису зашифрованого файлу
void encryptFile(const char* inputFile, const char* outputFile) {
    FILE* print = fopen(inputFile, "rb");
    if (!print) {
        cout << "Error while opening the file " << inputFile << endl;
    }
}

```

```

        return;
    }

    char buffer_for_original[256];
    cout << "Your file:\n";
    while (fgets(buffer_for_original, sizeof(buffer_for_original),
print)) {
        cout << buffer_for_original;
    }
    cout << endl;
    fclose(print);

    FILE* input = fopen(inputFile, "rb");
    if (!input) {
        cout << "Error while opening the file " << inputFile << endl;
        return;
    }
    FILE* output = fopen(outputFile, "wb");
    if (!output) {
        cout << "Error while opening the file " << outputFile <<
endl;
        return;
    }

    char buffer[256];
    cout << "\nYour encrypted file:\n";
    while (fgets(buffer, sizeof(buffer), input)) {
        for (size_t i = 0; i < strlen(buffer); i++) {
            if (buffer[i] != '\n' && buffer[i] != '\r') {
                buffer[i] = buffer[i] + 1;
            }
        }
        cout << buffer;
        fputs(buffer, output);
    }
    fclose(input);
    fclose(output);
}

//файл для дешифрування
void decryptFile(const char* inputFile, const char* outputFile) {
    FILE* print = fopen(inputFile, "rb");

```

```

    if (!print) {
        cout << "Error while opening the file " << inputFile << endl;
        return;
    }

    char buffer_for_original[256];
    cout << "Your file:\n";
    while (fgets(buffer_for_original, sizeof(buffer_for_original),
print)) {
        cout << buffer_for_original;
    }
    cout << endl;
    fclose(print);

    FILE* input = fopen(inputFile, "rb");
    if (!input) {
        cout << "Error while opening the file " << inputFile << endl;
        return;
    }
    FILE* output = fopen(outputFile, "wb");
    if (!output) {
        cout << "Error while opening the file " << outputFile <<
endl;
        return;
    }

    char buffer[256];
    cout << "\nYour decrypted file:\n";
    while (fgets(buffer, sizeof(buffer), input)) {
        for (size_t i = 0; i < strlen(buffer); i++) {
            if (buffer[i] != '\n' && buffer[i] != '\r') {
                buffer[i] = buffer[i] - 1;
            }
        }
        cout << buffer;
        fputs(buffer, output);
    }
    fclose(input);
    fclose(output);
}

```

```

int main() {
    const char* inputFile = "input.txt";
    const char* encryptedFile = "encrypted.txt";
    const char* decryptedFile = "decrypted.txt";
    int input = 0;
    //меню для роботи з програмою
    Menu:
    cout << "Welcome to spyware, how can I help you:\n";
    cout << "Enter \"1\" to encrypt the file\n";
    cout << "Enter \"2\" to decrypt the file\n";
    cout << "Press \"0\" to exit the program\n";
    cin >> input;
    switch(input){
        case 0:
            cout << "Goodbye";
            return 0;
        case 1:
            encryptFile(inputFile, encryptedFile);
            break;
        case 2:
            decryptFile(encryptedFile, decryptedFile);
            break;
        default:
            cout << "Wrong option\n";
            goto Menu;
    };

    return 0;
}

```

5)Результати виконання завдань та фактично затрачений час

John Black - Epic 5 Task 3 - Lab# programming: VNS Lab 6(варіант 4)

```
PS D:\c++\epic5> & 'c:\Users\User\.vscode\extensions\ne-In-ia5jxodd.pku' '--stdout=Microsoft-MIEngine-Out-b  
'--dbgExe=C:\msys64\mingw64\bin\gdb.exe' '--interprete  
apple banana cherry strawberry apple apple  
The first word: apple  
apple  
apple  
PS D:\c++\epic5> █
```

Фактичний час виконання: 24хв

John Black - Epic 5 Task 4 - Lab# programming: VNS Lab 8(варіант 4)

Menu:

1. Add a person
2. Print file contents
3. Delete records by age
4. Add a record after a specific phone number
5. Exit

Choose an option: 1

Enter details of the person:

Surname: Klymchuk

Name: Yurii

Patronymic: Olehovich

Address: Sunstreet55b

Phone: 0508192422

Age: 17

Person added successfully.

Menu:

1. Add a person
2. Print file contents
3. Delete records by age
4. Add a record after a specific phone number
5. Exit

Choose an option: 2

File contents:

Surname: Klymchuk, Name: Yurii, Patronymic: Olehovich

Address: Sunstreet55b, Phone: 0508192422, Age: 17

Menu:

1. Add a person
2. Print file contents
3. Delete records by age
4. Add a record after a specific phone number
5. Exit

Choose an option: 4

Enter phone number: 0508192422

Enter details of the new person:

```
Surname: Ivaniv
Name: Igor
Patronymic: Petrovych
Address: Lynkol`s13
Phone: 044253313
Age: 16
New record added after the record with phone number 0508192422.
```

```
Menu:
```

1. Add a person
2. Print file contents
3. Delete records by age
4. Add a record after a specific phone number
5. Exit

```
Choose an option: 3
```

```
Enter age to delete records: 16
```

```
Records with age 16 have been deleted.
```

```
Menu:
```

1. Add a person
2. Print file contents
3. Delete records by age
4. Add a record after a specific phone number
5. Exit

```
Choose an option: 2
```

```
File contents:
```

```
Surname: Klymchuk, Name: Yurii, Patronymic: Olehovych
Address: Sunstreet55b, Phone: 0508192422, Age: 17
```

```
-----
```

```
Menu:
```

1. Add a person
2. Print file contents
3. Delete records by age
4. Add a record after a specific phone number
5. Exit

```
Choose an option: 5
```

Фактичний час виконання: 1год 38хв

John Black - Epic 5 Task 5 - Lab# programming: VNS Lab 9(варіант 4)


```

The number of symbols in last word of the second file is 3
PS D:\c++\epic5> & 'c:\Users\User\.vscode\extensions\ms-vscode.cpptools-1
ne-In-ahtqw4z1.1z0' '--stdout=Microsoft-MIEngine-Out-l24zo1rm.jed' '--std
'--dbgExe=C:\msys64\mingw64\bin\gdb.exe' '--interpreter=mi'
The number of symbols in last word of the second file is 3
PS D:\c++\epic5> & 'c:\Users\User\.vscode\extensions\ms-vscode.cpptools-1
ne-In-vdtjmwfx.w0f' '--stdout=Microsoft-MIEngine-Out-yjhscjc1.paj' '--std
'--dbgExe=C:\msys64\mingw64\bin\gdb.exe' '--interpreter=mi'
The number of symbols in last word of the second file is 3
PS D:\c++\epic5> & 'c:\Users\User\.vscode\extensions\ms-vscode.cpptools-1
ne-In-ryu52pla.jqr' '--stdout=Microsoft-MIEngine-Out-p5vbyw40.u11' '--std
'--dbgExe=C:\msys64\mingw64\bin\gdb.exe' '--interpreter=mi'
The number of symbols in last word of the second file is 3
PS D:\c++\epic5>

```

Фактичний час виконання: 58хв

John Black - Epic 5 Task 6 - Lab# programming: Algotester Lab 4(варіант 1 з використанням STL)

```

5
1 3 2 4 5
4
2 5 0 7
3
1 3 4
2
0 7
2
2 5
7
0 1 2 3 4 5 7
5
0 1 3 4 7

```

Фактичний час виконання: 1год 41хв

John Black - Epic 5 Task 6 - Lab# programming: Algotester Lab 4(варіант 1 з власною імплементацією)

```
5 13 122 5 13 6655678882
```

```
7
4 3 8 4 6 0 5
3
2 9 3
6
0 4 4 5 6 8
2
2 9
1
3
9
0 2 3 4 4 5 6 8 9
8
0 2 4 4 5 6 8 9
```

Фактичний час виконання: 2год 11хв

John Black - Epic 5 Task 7 - Lab# programming: Algotester Lab 6(варіант 1)

```
5 2
stugna
neptune
grim
oplot
Grim
4
r m i g
```

Фактичний час виконання: 1год 1хв

John Black - Epic 5 Task 8 - Practice# programming: Class Practice Task(завдання 1)

```
ne-In-3kccujy5.bok' '--stdout=Microsoft-MIEngine-Out-3hr1lwjf.mbv' '--stderr=Microsoft
'--dbgExe=C:\msys64\mingw64\bin\gdb.exe' '--interpreter=mi'
Enter the file name: gaga.txt
Enter the content of the file: sbsh shbhdshf bsfbsh sbshb
The operation is successful
```

Фактичний час виконання: 32хв

John Black - Epic 5 Task 8 - Practice# programming: Class Practice Task(завдання 2)

```
'--dbgExe=C:\msys64\mingw64\bin\gdb.exe' '--interpreter=mi'  
Error while closing the file file_to.txt  
The operation failed  
PS D:\c++\epic5> & 'c:\Users\User\.vscode\extensions\ms-vscode.cpptools-1.2  
ne-In-lzfw1je0.ng1' '--stdout=Microsoft-MIEngine-Out-0waolsji.4co' '--stderr  
'--dbgExe=C:\msys64\mingw64\bin\gdb.exe' '--interpreter=mi'  
The operation is successful  
PS D:\c++\epic5> █
```

Фактичний час виконання: 23хв

John Black - Epic 5 Task 9 - Practice# programming: Self Practice Task

```
Welcome to spyware, how can I help you:  
Enter "1" to encrypt the file  
Enter "2" to decrypt the file  
Press "0" to exit the program  
1  
Your file:  
When I find myself in times of trouble  
Mother Mary comes to me  
Speaking words of wisdom  
Let it be  
And in my hour of darkness  
She is standing right in front of me  
Speaking words of wisdom  
Let it be  
  
Your encrypted file:  
Xifo!J!gjoe!nztfmj!jo!ujnft!pg!uspvcmf  
Npuijs!Nbsz!dpnft!up!nf  
Tqfbljoh!xpset!pg!xjtep  
Mfu!ju!cf  
Boe!jo!nz!ipvs!pg!ebsloftt  
Tif!jt!tuboejoh!sjhiu!jo!gspou!pg!nf  
Tqfbljoh!xpset!pg!xjtep  
Mfu!ju!cf
```

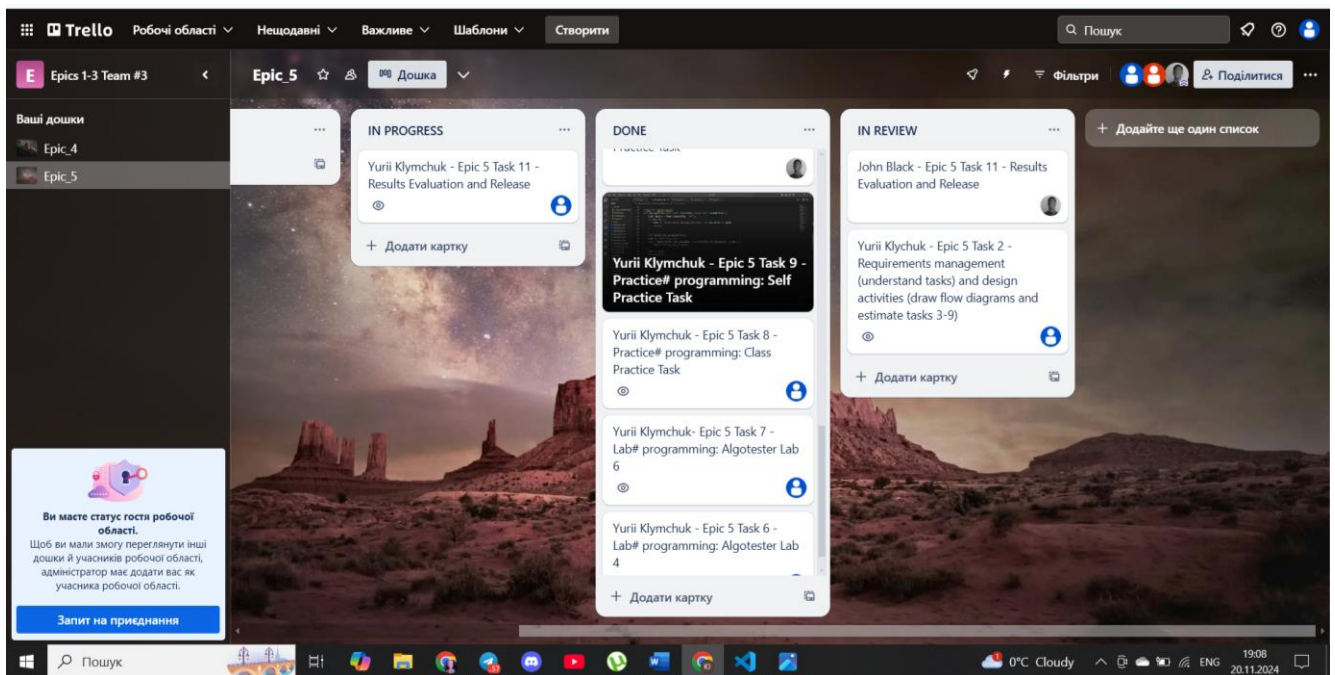
```
--dbg.exe-C:\msys64\mingw64\bin\gdb.exe --interpreter=mi
Welcome to spyware, how can I help you:
Enter "1" to encrypt the file
Enter "2" to decrypt the file
Press "0" to exit the program
2
Your file:
Xifo!J!gjoe!nztfmg!jo!ujnft!pg!uspvcmf
Npuifs!Nbsz!dpnft!up!nf
Tqfbljoh!xpset!pg!xjtepn
Mfu!ju!cf
Boe!jo!nz!ipvs!pg!ebsloftt
Tif!jt!tuboejoh!sjhiu!jo!gspou!pg!nf
Tqfbljoh!xpset!pg!xjtepn
Mfu!ju!cf

Your decrypted file:
When I find myself in times of trouble
Mother Mary comes to me
Speaking words of wisdom
Let it be
And in my hour of darkness
She is standing right in front of me
Speaking words of wisdom
Let it be
```

Фактичний час виконання: 1год 29хв

6)Робота з комадою

Trello:



Висновок: Робота з файлами, як текстовими, так і бінарними, є важливим елементом програмування, що дозволяє зберігати, обробляти та передавати дані. Стандартна бібліотека надає зручний набір інструментів для роботи з файлами, які спрощують операції читання, запису та обробки даних. Володіння цими навичками є важливим для створення ефективного та масштабованого програмного забезпечення, а також для розробки власних бібліотек, що адаптовані до специфічних потреб.

Посилання на пул реквест: https://github.com/artificial-intelligence-department/ai_programming_playground_2024/pull/397