



ĐẠI HỌC BÁCH KHOA HÀ NỘI

TRƯỜNG ĐIỆN-ĐIỆN TỬ

Cấu trúc dữ liệu và giải thuật

BÀI TẬP LỚN

GỢI Ý TÌM BẠN TRÊN MẠNG XÃ HỘI

i Giảng viên hướng dẫn: Trần Thị Thanh Hải

GIỚI THIỆU BÀI TOÁN: TẠO MẠNG XÃ HỘI CỦA LỚP

INPUT & OUTPUT	<ul style="list-style-type: none">Nhận thông tin từ người dùng như thông tin cá nhân, sở thích.Phản hồi bằng cách hiển thị nội dung cá nhân hóa, thông báo hoặc kết quả tương tác theo thời gian thực.
Động lực thực hiện	<ul style="list-style-type: none">Tạo 1 môi trường để những người có chung một số tiêu chí dễ dàng giao lưu và trao đổi.Học hỏi thêm kinh nghiệm trong quá trình làm bài tập lớn cùng nhau.
Ứng dụng trong thực tế	<ul style="list-style-type: none">Giúp sinh viên trao đổi thông tin với nhau qua mạng xã hộiXa hơn mô hình mạng xã hội nhỏ từ đó có thêm nhiều hướng phát triển cho một mạng xã hội lớn hơn.

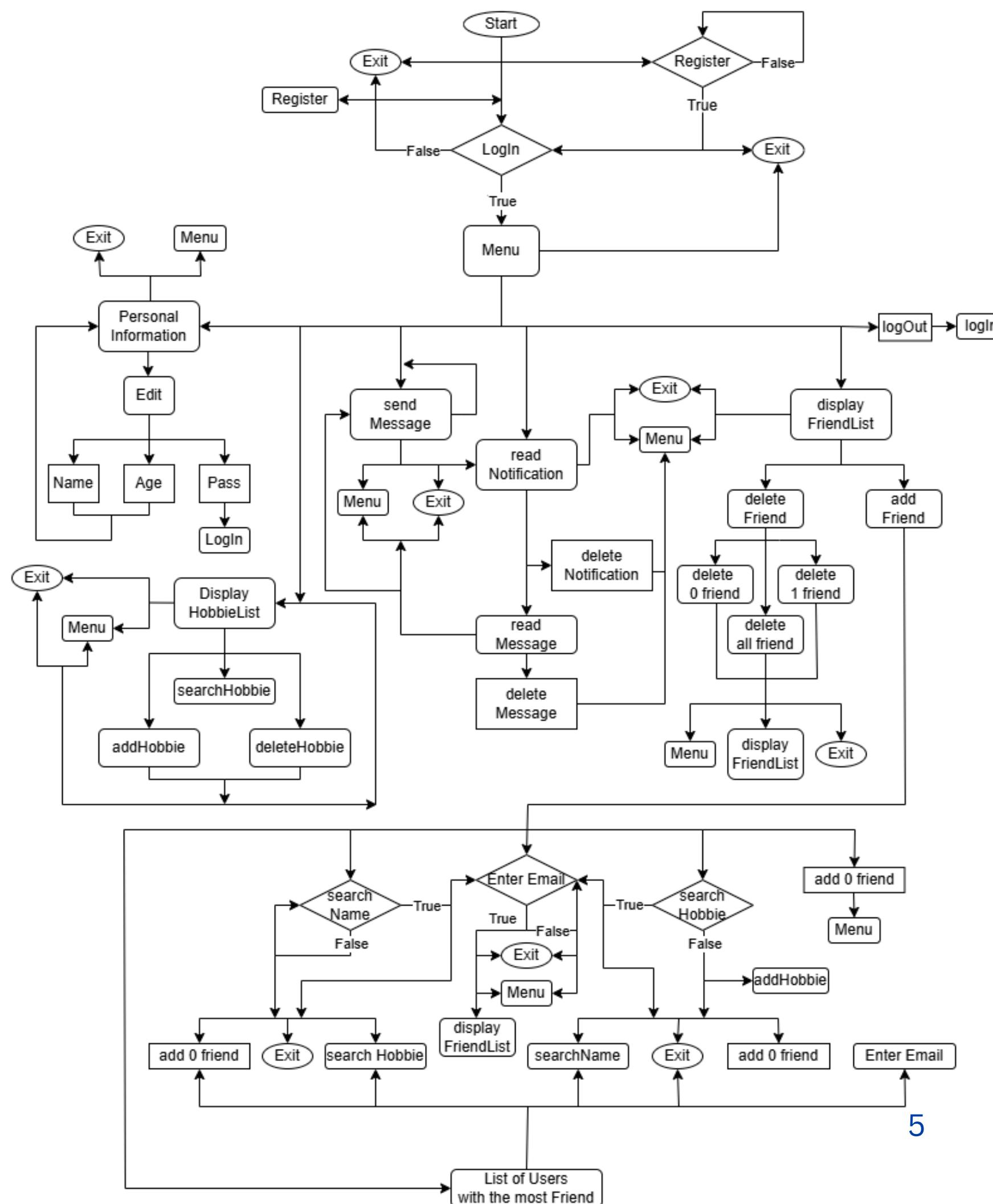
GIỚI THIỆU THÀNH VIÊN

Họ và tên	NHIỆM VỤ	ĐÁNH GIÁ
Ngô Thị Quỳnh Anh 20233250	<ul style="list-style-type: none">Tìm hiểu đề tàiLên ý tưởng chính, vẽ luồng chạy codeTriển khai và cài đặt chương trình-Coder chính, làm video demoHướng dẫn và phổ biến kiến thức chính cho cả nhóm	Đã hoàn thành
Nguyễn Hồng Phúc 20233581	<ul style="list-style-type: none">Tìm hiểu đề tàiTìm hiểu về Lưu trữ dữ liệuTriển khai và cài đặt chương trình-Coder phụPhát triển sản phẩm	Đã hoàn thành
Hứa Vũ Kiệt 20233481	<ul style="list-style-type: none">Tìm hiểu đề tàiTìm hiểu về Xử lý ngoại lệTriển khai và cài đặt chương trình-Coder phụPhát triển sản phẩm	Đã hoàn thành
Nguyễn Trung Nam 20233547	<ul style="list-style-type: none">Tìm hiểu đề tàiTìm hiểu về Cấu trúc dữ liệuLàm PowerPoint giới thiệuPhát triển sản phẩm	Đã hoàn thành

A. GIỚI THIỆU CHUNG

1. Lập trình 1 mạng xã hội đơn giản với các tính năng chính như: đăng nhập, đăng ký, kết bạn, tìm bạn theo sở thích
2. Sử dụng phương pháp lập trình cấu trúc thủ tục (kết hợp CTDL> để giải quyết bài toán)
3. Ngôn ngữ lập trình C++
4. Lập trình dòng lệnh và chạy bằng console

Luồng chạy chương trình

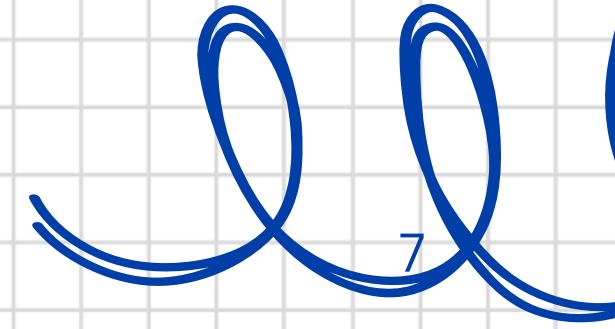
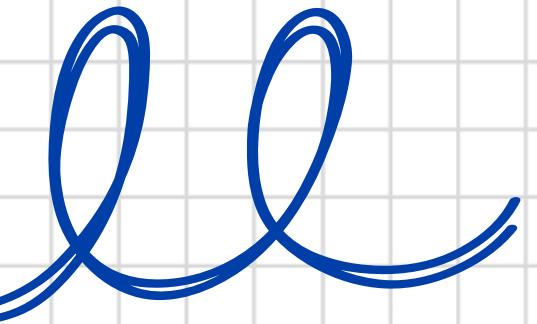


B. Phân tích kỹ thuật

I. LƯU TRỮ DỮ LIỆU	II. CẤU TRÚC DỮ LIỆU	III. GIẢI THUẬT	IV. XỬ LÝ NGOẠI LỆ
Sử dụng file text để lưu trữ dữ liệu	<ol style="list-style-type: none">Struct UserVectorMap và MultimapStringCác Cấu trúc dữ liệu tệp	<ol style="list-style-type: none">Xử lý fileTìm kiếm và duyệtXóa phần tửXử lý chuỗiSắp xếpLấy thời gian thựcXử lý thông báo và tin nhắnĐăng ký và Đăng nhập	<ol style="list-style-type: none">Xử lý khi nhập lựa chọnXử lý lỗi khi nhập đầu vàoXử lý ngoại lệ trong nhập thông tin cá nhânXử lý lỗi nghiêm trọng

*

I. Lưu trữ dữ liệu



Social_Network



Friends / Hobbies / Notification / Users

Friends

Hobbies

Messages

Notification

Users

List_of_available_preferences.txt

Number_of_Accounts.txt

Social_Network.cpp

Social_Network.exe

Social_Network.o



1.txt

2.txt

3.txt

4.txt



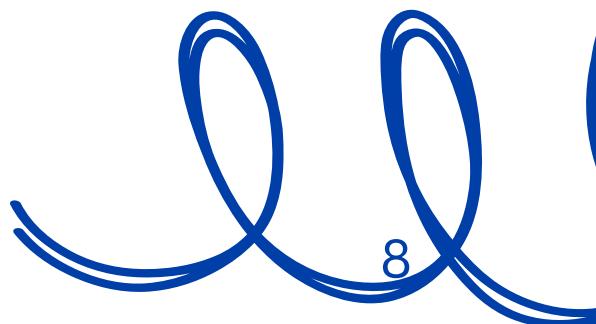
Messages

5_4.txt

11_2.txt

2_3.txt

2_1.txt





Friends\\2.txt

ntqa@gmail.com

nguyenquang5625@gmail.com

kiet.hv@gmail.com

chaudt07@gmail.com



Users\\2.txt

2

Ngo Thi Quynh Anh

19

ntqa@gmail.com

123456



Hobbies\\2.txt

ntqa@gmail.com

Nghe nhac

Nau an

Tap gym

Nuoi thu cung



Notification\\5.txt

Mon Dec 16 08:33:30 2024

Chao mung den voi Social Network!

Fri Dec 16 09:05:06 2024

dat.bt@gmail.com da ket ban voi ban!



List_of_available_preferences.txt

Choi game

Chup anh

Nghe nhac

Doc sach

Xem phim

Choi the thao

Choi nhac cu

Tham gia mang xa hoi



Messages\\2_1.txt

Thu Dec 12 09:37:50 2024

Message sent by Ngo Thi Quynh Anh:
xin chao!

<new message>

Thu Dec 12 09:39:05 2024

Message sent by Nguyen Quang Son:
Hello World!



Number_of_Accounts.txt

10

*

II. Cấu trúc dữ liệu

1. Struct User

- Dùng để đại diện thông tin người dùng:
- userID (long long): ID của người dùng.
- userName (string): Tên người dùng.
- userAge (int): Tuổi của người dùng.
- userEmail (string): Email đăng nhập.
- password (string): Mật khẩu.



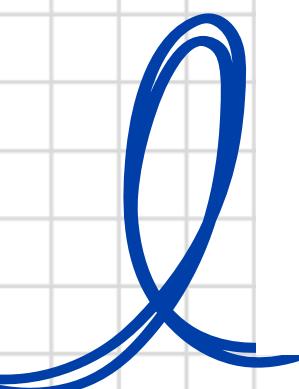
2. Vector

- Lưu danh sách tất cả người dùng (vector<User>)
- Lưu danh sách bạn bè, danh sách sở thích của người dùng cụ thể (vector<string>)
- Lưu danh sách sở thích có sẵn (vector<string>)
- Lưu danh sách các sở thích bạn muốn xóa (vector<string>)
- Lưu từng dòng tin nhắn (vector<string>)



3. Map và Multimap

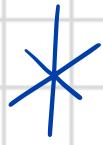
- map<string, vector<string>> : Lưu danh sách bạn bè, danh sách sở thích của tất cả người dùng dưới dạng <user, userList>
- multimap<int, User> : sắp xếp người dùng theo thứ tự để + tìm kiếm theo tên (sắp xếp người dùng có tên chứa nhiều ký tự chung nhất với tên mình tìm kiếm)
+ tìm kiếm theo sở thích (sắp xếp người dùng có nhiều sở thích chung với mình nhất)
- multimap<int, string> : sắp xếp tất cả người dùng theo số lượng bạn bè của người đó



4.String

Lưu và xử lý dữ liệu văn bản như: tên người dùng, email, mật khẩu, nội dung thông báo, nội dung tin nhắn, ...





5. Các CSDL tệp

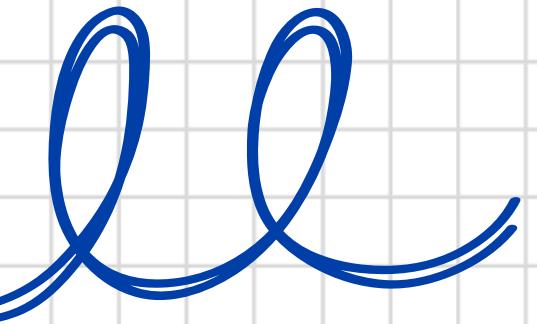
Lưu trữ và quản lý thông tin:

- + Người dùng: Thông tin cá nhân, bạn bè, sở thích
- + Danh sách sở thích có sẵn (file List_of_available_preferences.txt)
- + Tin nhắn (Messages\\<ID1>_<ID2>.txt)
- + Thông báo (Notification\\<ID>.txt)





III. Giải thuật



1. Xử lý file

- Đọc file (ifstream)

VD: Hàm getUser() : đọc danh sách tất cả người dùng

Hàm getList() : đọc danh sách bạn bè hoặc danh sách sở thích của tất cả người dùng

Hàm getYourList() : đọc danh sách bạn bè hoặc danh sách sở thích của người dùng cụ thể

- Ghi file (ofstream)

VD: hàm sendNotification() : ghi thông báo vào file

- Kiểm tra file tồn tại

VD: hàm isFileExist() : dùng ifstream để kiểm tra sự tồn tại của file

- Kiểm tra file rỗng

VD: hàm isEmptyFile() : dùng ifstream kiểm tra file có rỗng không



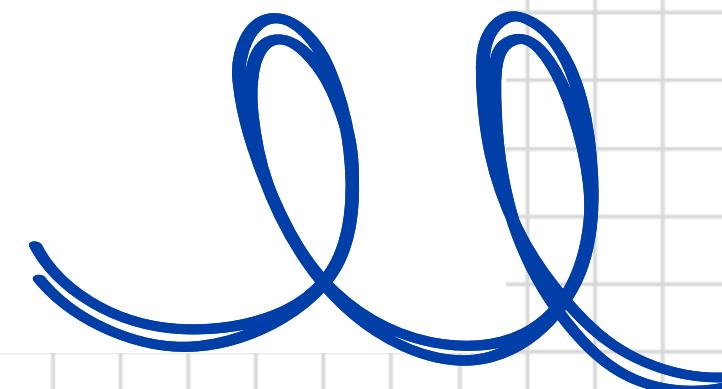
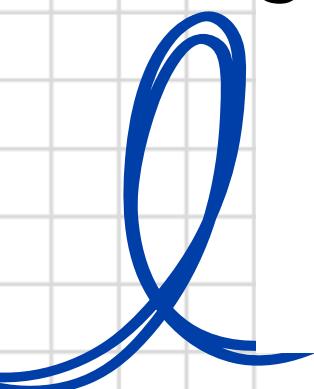
2. Tìm kiếm và duyệt

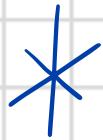
- **Tìm kiếm tuần tự:** dùng `find` để kiểm tra một sở thích hay email có tồn tại trong vector không

VD: `isHobbiesExist()`, `isFriendExist()`

- **Duyệt danh sách:** duyệt qua danh sách người dùng, sở thích, thông báo, ...

VD: hàm `searchHobbie()` duyệt tất cả người dùng để tìm người có sở thích chung

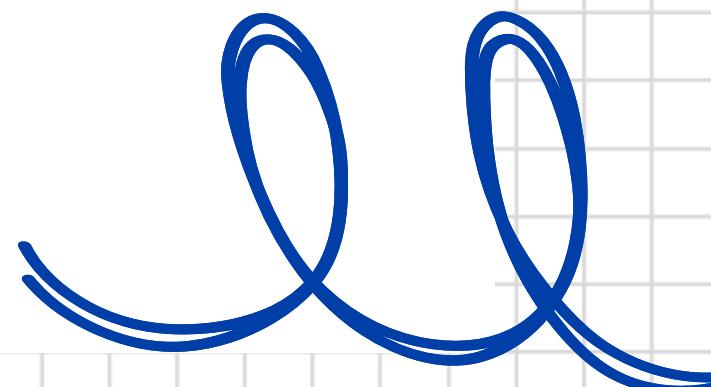
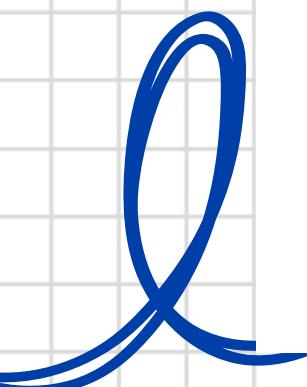




3. Xóa phần tử

-Sử dụng **remove** kết hợp với **vector.erase()** để xóa phần tử khỏi vector

VD: Chương trình sẽ duyệt qua danh sách sở thích bạn muốn xóa và dùng **remove** kết hợp với **vector.erase()** để xóa phần tử ra khỏi vector sở thích, sau đó cập nhật những sở thích sau khi xóa vào file bằng ofs out



* 4. Xử lý chuỗi

- Chuyển đổi chữ hoa chữ thường: sử dụng `transform()` để chuyển chuỗi về chữ hoa hoặc chữ thường trước khi so sánh

- So sánh chuỗi

VD: hàm `So_ky_tu_giong_nhau_lien_tiep_lon_nhat()` tìm số ký tự liên tiếp giống nhau trong 2 chuỗi để thực hiện tìm kiếm theo tên

- Ghép chuỗi

VD: `"Messages\" + to_string(getID(senderEmail)) + "_" + to_string(getID(receiverEmail)) + ".txt"`: ghép chuỗi để tạo tên file



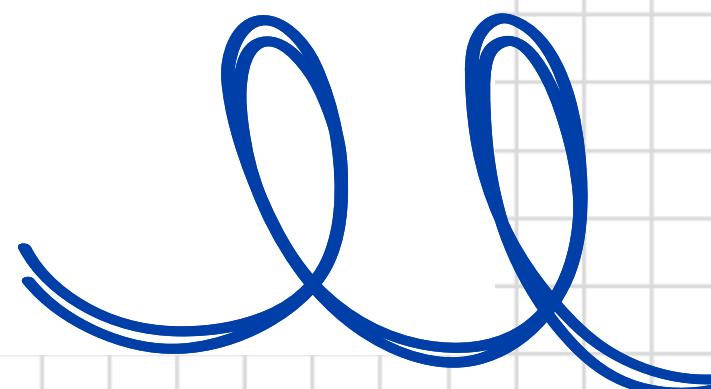
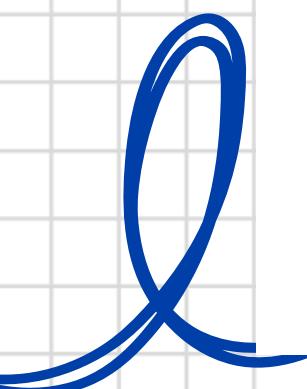
5. Sắp xếp

- Multimap sắp xếp tự động

VD: **multimap<int, User>** trong hàm **searchHobbie()**: thì nó sẽ lưu dưới dạng số sở thích chung của User và User và sắp xếp theo tăng dần của key, sau đó đảo ngược multimap để in ra màn hình

VD: **multimap<int, string>** trong

Danh_sach_Users_sap_xep_theo_So_ham(): int ra màn hình lần lượt từ người có số lượng bạn lớn nhất sau đó giảm dần

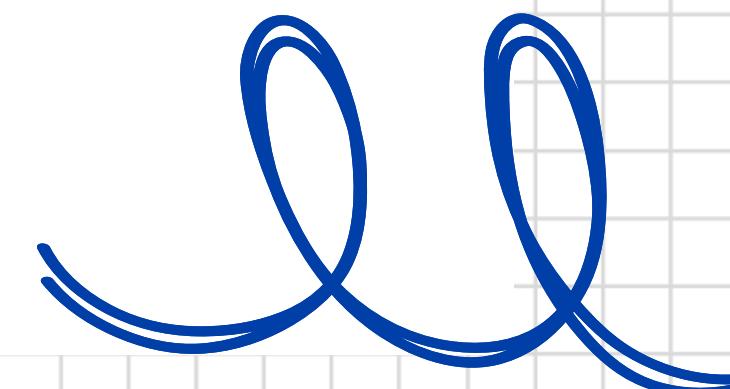
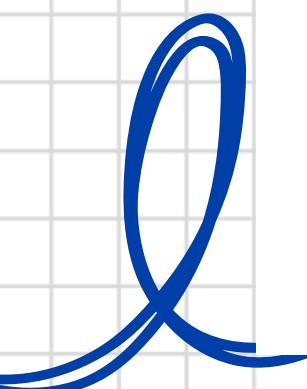




6. Lấy thời gian thực:

Hàm **getTime()** thực hiện các bước sau:

- Sử dụng **time()** để lấy dấu thời gian hiện tại (timestamp) từ hệ thống.
- Biến đổi dấu thời gian thành chuỗi thời gian có định dạng dễ đọc (như Thu Dec 23 14:55:02 2024\n) thông qua hàm **ctime()**.
- Trả về chuỗi kết quả.



7. Xử lý thông báo và tin nhắn

- **Gửi thông báo:** Ghi thông báo vào file dựa trên ID người nhận
- **Gửi tin nhắn:** khi mình gửi tin nhắn thì từng dòng tin nhắn sẽ lưu vào vector và sau đó
 - nếu trong tin nhắn chưa có xâu <new message> thì thêm vào.
 - mình sẽ lưu tin nhắn vào file(ofs append)
- **Đọc tin nhắn:** sử dụng ifstream để đọc file từ filetext, sau đó lưu vào vector.Nếu mà có <new message>, mình sẽ xóa nó đi và lưu vào file.

8. Đăng nhập và Đăng ký

- **Kiểm tra thông tin đăng nhập:** Hàm `isCorrectLoginInfo()` so sánh mật khẩu và email với dữ liệu trong file
- **Tạo tài khoản mới:** Hàm `registerAccount()` lưu thông tin người dùng mới vào file mới

*

IV. Xử lý ngoại lệ

1.Xử lý lỗi khi nhập lựa chọn

- **Lỗi mắc phải:** nhập lựa chọn nằm ngoài miền giá trị
- **Cách xử lý:**
 - + Khi nhập lựa chọn trong hàm **choice()**: hiện thị thông báo lỗi và yêu cầu nhập lại
 - + Khi thêm hoặc xóa sở thích: bỏ qua lựa chọn lỗi



2.Xử lý lỗi khi nhập đầu vào

- Lỗi mắng phải: nhập đầu vào không phải số nguyên
- Cách xử lý: Nếu đầu vào lỗi
 - + Xóa cờ lỗi của cin: `cin.clear()`
 - + Bỏ qua phần dữ liệu còn lại trong dòng:
`cin.ignore(numeric_limits<streamsize>::max(), '\n')`
 - + Hiển thị thông báo lỗi và yêu cầu người dùng nhập lại

3.Xử lý ngoại lệ khi nhập thông tin cá nhân

a) Kiểm tra tên người dùng

- **Lỗi mắc phải:** nhập tên người dùng không có ký tự hoặc chỉ chứa các ký tự space
- **Cách khắc phục:**
 - + Xóa toàn bộ khoảng trắng trong tên để kiểm tra xem người dùng có nhập toàn khoảng trắng hay không
 - + Nếu tên không hợp lệ hiển thị thông báo và yêu cầu nhập lại

3.Xử lý ngoại lệ khi nhập thông tin cá nhân

b) Kiểm tra tuổi người dùng

- **Lỗi mắc phải:** nhập tuổi không dương hoặc không phải số nguyên
- **Cách khắc phục:** nếu tuổi không hợp lệ hiển thị thông báo và yêu cầu nhập lại

c) Kiểm tra mật khẩu

- **Lỗi mắc phải:** Nhập mật khẩu ngắn hơn 6 ký tự
- **Cách khắc phục:** nếu mật khẩu không hợp lệ hiển thị thông báo và yêu cầu nhập lại

4. Xử lý lỗi nghiêm trọng.

- **Lỗi mắc phải:** Không thể mở file hoặc file không tồn tại
- Không thể khắc phục, chương trình dừng ngay lập tức `exit(0)`

***Tổng hợp: Kỹ thuật xử lý ngoại lệ trong chương trình tập trung vào:**

- Bảo vệ hệ thống khỏi các lỗi nhập sai của người dùng
- Kiểm tra trạng thái file và đảm bảo hoạt động đọc, ghi không gặp lỗi
- Thoát chương trình khi gặp lỗi nghiêm trọng (như không tìm thấy file)

Các khó khăn gặp phải khi triển khai và cách giải quyết thế nào

- Thiết kế giao diện phù hợp với tất cả người dùng
--> Tạo giao diện đơn giản dễ ai cx có thể sử dụng được
- Bảo mật thông tin của của người dùng

Hạn chế còn tồn tại và hướng phát triển

- Chưa có nhiều tính năng nâng cao
- Giao diện chưa tối ưu
- Hướng phát triển : có thêm tính năng điểm danh tự động

Môi trường lập trình, các thư viện sử dụng

- Sử dụng thư viện chuẩn `<bits/stdc++.h>` để truy cập toàn bộ thư viện C++ tiêu chuẩn.
- Các thư viện quan trọng được sử dụng bao gồm: `<iostream>`, `<fstream>`, `<vector>`, `<string>`, `<map>`, và các chức năng liên quan đến thời gian (`<ctime>`).
- Chương trình này được thiết kế để chạy trong một môi trường dòng lệnh