TRƯỜNG ĐẠI HỌC XÂY DỰNG HÀ NỘI

KHOA CÔNG NGHỆ THÔNG TIN

BÀI TẬP LỚN

# Xây dựng chương trình quản lí việc đăng ký môn học của sinh viên một khoa có N sinh viên, M môn học gồm những chức năng: nhập, xem toàn bộ, tìm kiếm và thống kê. Trong đó tiền học được tính theo số tín chỉ mà sinh viên đăng ký

Sinh viên thực hiện:

- Nguyễn Tiến Hưng - 1644169 - 69CS1

- Đinh Trọng Anh - 0364369 - 69CS1

- Nghiêm Trường An - 0364069 - 69CS1

- Hà Nguyễn Minh Hiếu - 0367669 - 69CS1

Hà Nội, tháng 5 năm 2025

**Chương 1: Mô tả chương trình**

- Chương trình được chia thành nhiều file: main.cpp, functions.cpp, và file header

- Các chức năng chính gồm: nhập dữ liệu, tìm kiếm, liệt kê danh sách sinh viên/môn học, và ghi dữ liệu ra file.

- Thành viên thực hiện:

+ Thiết kế file header, menu, hàm check mã sinh viên và mã môn học, slide, báo cáo: Nguyễn Tiến Hưng (nhóm trưởng)

+ Thống kê, hiển thị danh sách: Đinh Trọng Anh

+ Tìm kiếm, slide, báo cáo: Nghiêm Trường An

+ Viết hàm nhập dữ liệu, sơ đồ khối, check tên sinh viên và tín chỉ: Hà Nguyễn Minh Hiếu

## 1.1 Mô tả hàm main()

-Hàm main là trung tâm điều khiển chính của chương trình, thực hiện các công việc như khởi tạo dữ liệu, đọc dữ liệu từ tệp, hiển thị menu cho người dùng lựa chọn, và xử lý các chức năng tương ứng với từng lựa chọn. Cấu trúc và vai trò của từng phần trong hàm main như sau:

+)Khởi tạo vector chứa dữ liệu:



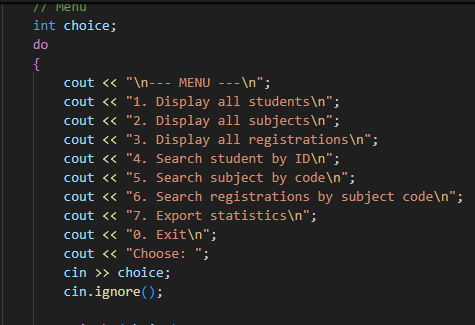
* Ba vector này lần lượt lưu trữ danh sách sinh viên, danh sách môn học và danh sách đăng ký môn học

+)Đọc dữ liệu từ tệp:



* Hàm này đọc dữ liệu từ các tệp văn bản tương ứng và lưu vào các vector. Dữ liệu bao gồm thông tin sinh viên, môn học và danh sách sinh viên đã đăng ký môn học.

+)Hiển thị menu và xử lý lựa chọn:





* Vòng lặp do...while đảm bảo chương trình luôn hiển thị menu cho đến khi người dùng chọn 0 để thoát.
* Người dùng nhập lựa chọn, sau đó chương trình sẽ gọi hàm tương ứng để thực hiện chức năng:

|  |  |
| --- | --- |
| Lựa chọn | Chức năng |
| 1 | Hiển thị toàn bộ danh sách sinh viên |
| 2 | |  | | --- | |  |  |  | | --- | | Hiển thị toàn bộ danh sách môn học | |
| 3 | Hiển thị toàn bộ danh sách đăng ký |
| 4 | Tìm kiếm sinh viên theo mã số |
| 5 | Tìm kiếm môn học theo mã môn |
| 6 | Tìm kiếm danh sách đăng ký theo mã môn học |
| 7 | Xuất thống kê |
| 0 | Thoát chương trình |

+)Xử lý nhập sai:

* Nếu người dùng nhập lựa chọn không nằm trong danh sách từ 0 đến 7,

chương trình sẽ thông báo "Invalid choice!" và yêu cầu nhập lại.

## Đọc dữ liệu từ tệp văn bản:

* Mô tả**:** Hệ thống cho phép đọc dữ liệu từ tệp input.txt để lấy thông tin sinh viên, môn học và đăng ký học.
* Chi tiết xử lý:

+) Kiểm tra file có tồn tại và mở thành công không.

+) Phát hiện và cảnh báo nếu có dữ liệu trùng lặp (ID sinh viên hoặc mã đăng ký).

+) Lưu dữ liệu đã đọc vào các vector tương ứng: students, subjects, regists.

## Hiển thị danh sách sinh viên, môn học và đăng ký:

* **Mô tả:** Cho phép người dùng in ra màn hình danh sách tất cả sinh viên, môn học và các đăng ký học phần hiện có.
* **Chi tiết xử lý:**

+) Sử dụng các hàm displayStudents, displaySubjects, displayRegistrations.

+) Thông tin sinh viên gồm mã số và tên.

+) Thông tin môn học gồm mã môn, tên môn, và số tín chỉ.

+) Thông tin đăng ký gồm tên sinh viên và danh sách các môn đã đăng ký.

## Tìm kiếm sinh viên theo mã số

* **Mô tả:** Người dùng có thể nhập mã sinh viên và hệ thống sẽ hiển thị thông tin tương ứng nếu tìm thấy.
* **Chi tiết xử lý:**

+) Hàm searchStudent duyệt qua vector students để so sánh với ID nhập vào.

+) Nếu tìm thấy, hiển thị thông tin; nếu không, thông báo không tìm thấy.

## Tìm kiếm sinh viên theo mã môn học:

* **Mô tả:** Cho phép tìm kiếm thông tin một môn học cụ thể dựa trên mã môn do người dùng nhập.
* **Chi tiết xử lý:**

+) Hàm searchSubject so sánh mã môn nhập vào với các phần tử trong subjects.

+) In ra thông tin nếu tìm thấy, hoặc báo lỗi nếu không có.

## Tìm kiếm danh sách sinh viên đã đăng ký môn học:

* **Mô tả:** Hệ thống cho phép nhập mã môn học và liệt kê các sinh viên đã đăng ký môn đó.
* **Chi tiết xử lý:**

+) Hàm searchRegistrationsBySubjectCode duyệt qua danh sách regists để tìm những sinh viên có môn học trùng với mã nhập vào.

+) Kết quả là danh sách sinh viên đăng ký môn học đó.

## Xuất thống kê môn học:

* **Mô tả:** Tạo thống kê số sinh viên đăng ký từng môn học, kèm học phí ước tính, và ghi ra file output.txt.
* **Chi tiết xử lý:**

+) Hàm generateStatistics tính số lượt đăng ký từng môn học.

+) Mỗi sinh viên đăng ký môn học sẽ được tính học phí là số tín chỉ \* 300.000 VNĐ.

+) Ghi thống kê vào file dưới dạng: Mã môn - Tên môn: số sinh viên - Tổng học phí.

Chương trình quản lý sinh viên thực hiện các công việc chính sau, được thể hiện qua các nhóm hàm sau đây:

1. **Đọc dữ liệu từ file**:

* readAllDataFromFile(): Hàm tổng hợp gọi các hàm con để đọc dữ liệu từ file.
* inputStudent(), inputSubject(), inputRegistration(): Đọc từng dòng dữ liệu từ file vào từng cấu trúc.

1. **Hiển thị danh sách**:

* displayStudents(), displaySubjects(), displayRegistrations(): In ra danh sách sinh viên, môn học, và đăng ký.

1. **Tìm kiếm thông tin**:

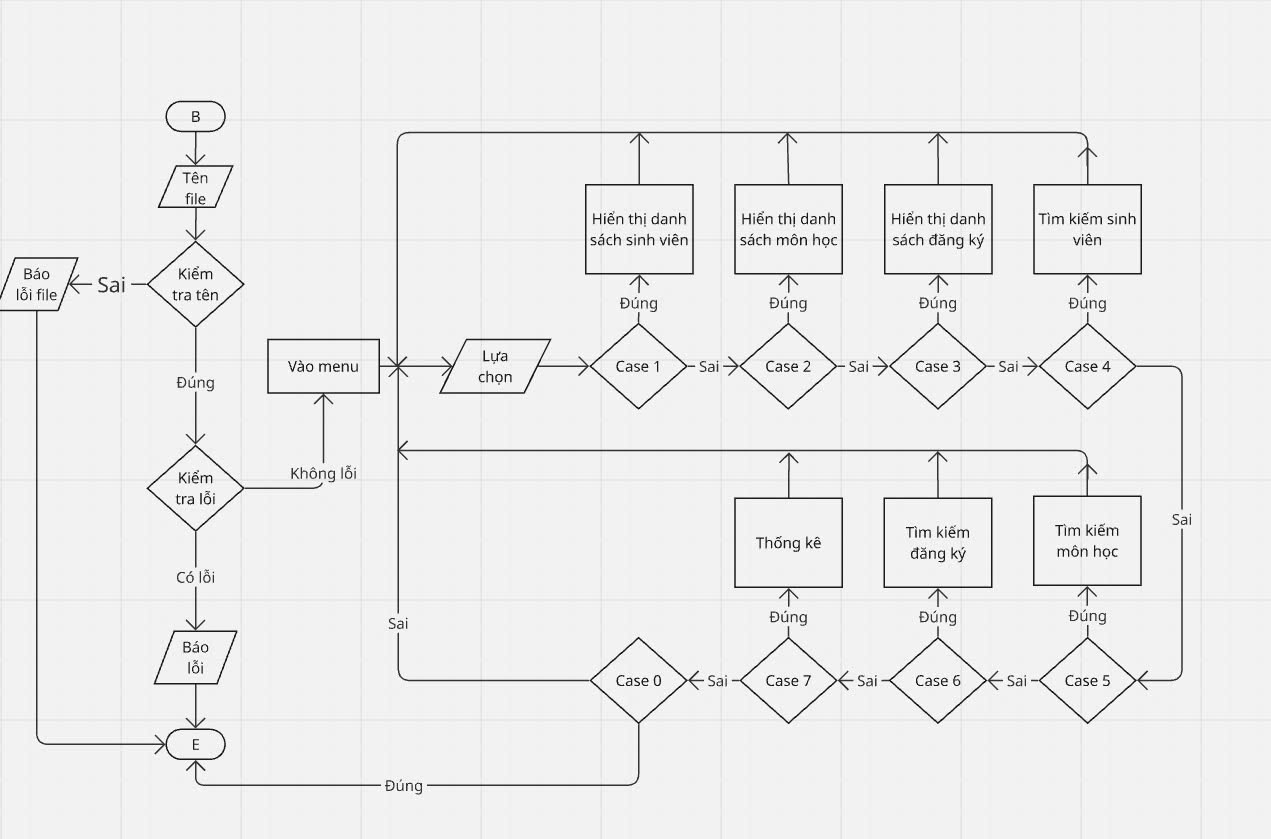
* searchStudent(): Tìm kiếm sinh viên theo ID.
* searchSubject(): Tìm kiếm môn học theo mã môn.
* searchRegistrationsBySubjectCode(): Tìm danh sách sinh viên đăng ký theo mã môn.

1. **Thống kê học phí**:

* generateStatistics(): Thống kê số sinh viên mỗi môn và tổng học phí, ghi ra file.
* formatVietnamese(): Định dạng tiền tệ kiểu Việt Nam (có dấu chấm).

Các chức năng trên được gọi trong hàm main() để tạo giao diện tương tác với người dùng.

* Sơ đồ khối của chương trình:



# Chương 2: Hoạt động của chương trình

Phần hoạt động này sẽ được chia thành các kịch bản :

Kịch bản 1: Những lỗi sai có thể gặp trong chương trình.

Kịch bản 2: Khi thao tác với dữ liệu đúng.

## 2.1 Kịch bản 1:

Khi thao tác với dữ liệu sai:

Mỗi một phần dữ liệu khi được nhập tử file input.txt đểu được kiểm tra rất rõ ràng:

-Lỗi ở phần sinh viên:

Có các hàm sử lý:

+if (s.checkDuplicateID(students, s.id)): Kiểm tra trùng lặp sinh viên

+if (!isValidStudentID(s.id)) :kiểm tra ID có sai định dạng (phải SVxx)

+if (!isValidName(s.name)) : kiểm tra tên không hợp lệ (chứa ký tự lạ)

-Lỗi ở phần môn học

+if (c.checkDuplicateCode(subjects, c.code)): Kiểm tra trùng mã môn

+if (c.checkDuplicateName(subjects, c.name)): Kiểm tra trùng tên môn

+if (!isValidSubjectCode(c.code)) : Kiểm tra mã môn sai định dạng (MHxx)

+if (!isValidSubjectCredit(c.credit)): Kiểm tra số tín chỉ sai (không phải số, hoặc <=0 hoặc >4)

+if (!isValidName(c.name)): Kiểm tra tên không hợp lệ

-Phần đăng kí môn học:

if (r.checkIdAndCode(students, subjects)): phần này kiểm tra hai điều:

+studentID có tồn tại trong danh sách sinh viên ?

+Mỗi subjectCode có tồn tại trong danh sách môn học?

Nếu bất kì phần nào sai trả về true ->báo lỗi

-Cách chương trình sử lý khi có lỗi:

A black background with green text

AI-generated content may be incorrect.

Nếu có bất kì lỗi nào chương trình sẽ thoát ngay lâp tức.

- Báo lỗi **cụ thể**, kèm **dữ liệu sai** → người dùng nhìn là biết file input đang sai chỗ nào do mỗi một hàm check đều in ra lỗi của hàm check đó giúp dễ bảo trì đảm bảo đầu vào luôn sạch sẽ

## 2.2 Kịch bản 2:

## 2.2.a Chuyển dữ liệu từ file input.txt vào các vector students, subjects, subjectCodes

+Mục đích: Đọc dữ liệu từ file input.txt và lưu vào các danh sách students, Subjects và reg để chương trình sử lý dữ liệu, đây là hàm quan trọng nhất của chương trình liên kết file lưu trữ trước đó và chương trình.

+Dữ liệu đầu vào: một file input.txt có định dạng như ảnh:

A screenshot of a computer

AI-generated content may be incorrect.

+Phân tích dữ liệu đầu vào:

File input.txt được chia thành 3 phần phần đầu là mã sinh viên và tên của học sinh, phần hai là phần của subject gồm mã môn, số tín chỉ và tên môn học, phần 3 là phần liên quan đến môn học sinh viên đăng kí gồm mã sinh viên và một chuỗi các môn học theo sau. Trước mỗi phần sẽ có đánh số để quy định số dòng cần đọc VD: số 3 là cần đọc 3 dòng.

+Các bước thực hiện:

\*Tổng quan: Hàm đọc thông tin từ file input.txt

A screenshot of a computer program

AI-generated content may be incorrect.

-Hàm sẽ thực hiện lần lượt các công việc sau:

1, Mở file có tên filename nếu không mở được → in lỗi và kết thúc hàm.

2, Đọc danh sách sinh viên: chương trình sẽ đọc số từ đầu file vd:n sau đó việc đọc file được lặp lại n lần mỗi một lần lặp sẽ đọc một sinh viên .

3, Đọc danh sách môn học: hoạt song với cơ chế tương tự phần 2 đọc lần lượt dữ liệu môn học và ghi vào vector.

4, Đọc mã sinh viên và mảng danh sách các sinh viên đã đăng kí: đọc n lần từ file mỗi một lần đọc từng dòng thêm dữ liệu vào vector.

-Đánh giá:

Ưu điểm:Việc tạo hàm đọc giúp rút ngắn việc đọc file tránh việc phải tạo 3 file, code sẽ ngắn gọ và dễ bảo trì và việc nhập dữ liệu vào file text cũng nhanh hơn, có tính đồng bộ dữ liệu cao.

Nhược điểm: bù lại File phức tạp, khó chỉnh sửa tay, khó mở rộng thêm dữ liệu và File dễ bị lỗi toàn cục mất định dạng dẫn đến mất dữ liệu nếu sai format.

-Để khắc phục bớt phần nhược điểm: hàm đã có các phần kiểm tra lỗi bị lặp bằng Kiểm tra trùng ID bằng các hàm s.checkDuplicate…→ Nếu trùng → Báo lỗi + Thoát chương trình (exit(1)) và yêu cầu nhập lại

## 2.2.b: phần Tìm kiếm

1,Phần tìm kiếm học sinh nhờ mã sinh viên

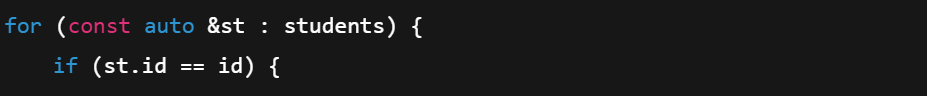
+Mục tiêu: Tìm kiếm sinh viên theo **Mã sinh viên (ID)** → Nếu có → In ra thông tin sinh viên + Danh sách môn học đã đăng ký + Tổng số tín chỉ.

+Các bước thực hiện của hàm:

B1: Nhập và kiểm tra mã sinh viên :

A computer screen with text

AI-generated content may be incorrect.

B2: Tìm kiếm sinh viên trong danh sách

B3:dùng để hiển thị thông tin sinh viên với đối tham số đầu vào là một học sinh hiển thị các thông tin như mã sinh viên và tên

A computer code with colorful text

AI-generated content may be incorrect.

-Hàm sẽ hiện thị “There is no student with ID” nếu không tìm thấy học sinh thỏa mãn

B4: Tính tổng tín chỉ + Hiển thị môn học đã đăng ký

Duyệt qua danh sách register tìm bản ghi đăng kí của sinh viên:

A computer screen with white text

AI-generated content may be incorrect.

Lặp qua môn học đã đăng kí của sinh viên hiển thị ra thông tin của sinh viên các môn học đăng kí của sinh viên đó đồng thời cộng thêm vào biến tổng tín chỉ

A screen shot of a computer code

AI-generated content may be incorrect.

Sau đó chương trình hiển thị ra tổng số tín chỉ của sinh viên đố đã đăng ký

2, Tìm kiếm môn học qua mã môn học

Với cơ chế tương tự với tìm kiếm sinh viên sử dụng tìm kiếm tuần tự, duyệt qua mảng subject được nhập nhờ hàm read data ở trên.

Hàm tìm kiếm môn học:

A screen shot of a computer program

AI-generated content may be incorrect.

3, Tìm và hiển thị danh sách sinh viên đã đăng ký một môn học cụ thể

Hàm nhận 3 danh sách (đều truyền bằng tham chiếu hằng để không sửa đổi và tránh sao chép):

**Bước 1:** Yêu cầu người dùng nhập mã môn học (code)

A black background with green text

AI-generated content may be incorrect.

**Bước 2:** Duyệt qua tất cả bản đăng ký (regists) để tìm **những sinh viên nào đã đăng ký môn học có mã = code**.

**A computer screen with white text

AI-generated content may be incorrect.**

**Bước 3: Nếu không ai đăng ký môn đó → in thông báo**

**Bước 4: Nếu có sinh viên → in danh sách sinh viên**

**A screen shot of a computer program

AI-generated content may be incorrect.**

## 2.2.3 Phần đưa ra báo cáo (output.txt)

Tổng quan:

A screen shot of a computer program

AI-generated content may be incorrect.

Hàm có tác dụng tạo báo cáo thống kê số lượng sinh viên đăng ký từng môn học và tổng học phí thu được của mỗi môn, sau đó ghi kết quả vào một file output.txt

-Ý tưởng chính:

→ Duyệt qua tất cả các môn học.  
→ Đếm số sinh viên đã đăng ký từng môn.  
→ Tính tổng học phí của môn đó.  
→ Ghi thống kê vào file (đường dẫn output.txt do người dùng truyền vào).

Phần chính của hàm là vòng lặp kép

Vòng lặp ngoài — duyệt từng môn học

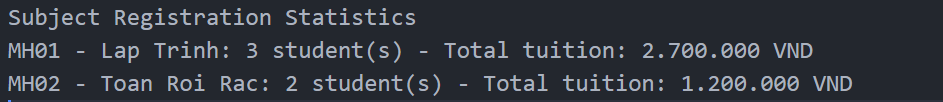
Vòng lặp trong — duyệt tất cả các bản đăng ký

Ý nghĩ: sử dụng hàm find và biến count:

Nếu sinh viên này đã đăng ký môn đang xét  
Tăng count thêm 1 (đếm thêm một sinh viên)

Từ biến count ta sẽ tính được tiền học phí thu được của môn đó

File output.txt sẽ được biễu diễn như sau:



# Chương 3: Kết luận

3.1 Những gì đạt được

+ Quản lý dữ liệu cơ bản: Đọc các danh sách đầu vào rồi hiển thị, tìm kiếm sinh viên, môn học và danh sách đăng ký

+ Kiểm tra lỗi đầu vào: Báo lỗi nếu trùng ID, mã môn học

+ Xuất thống kê ra file: Số lượng đăng ký 1 môn và tiền học

+ Tổ chức chương trình rõ ràng: Chia thành 1 file header, 1 file.cpp định nghĩa các hàm, 1 file main kèm theo 2 file .txt: 1 file số liệu đầu vào và 1 file kết quả đầu ra

3.2 Khả năng ứng dụng: Chương trình phù hợp cho trường học có quy mô nhỏ muốn quản lí danh sách sinh viên, môn học và sinh viên đã đăng ký môn học nào.

* 1. Những thứ còn tồn tại:

+ Chưa có giao diện người dùng thân thiện

+ Không có hệ thống quản lý người dùng: Không có hệ thống phân quyền người dùng(giảng viên, sinh viên, ...)

+ Không hỗ trợ đa ngôn ngữ hoặc Unicode

* 1. Khả năng phát triển và hướng cải thiện:

+ Xây dựng menu lựa chọn rõ ràng thay vì nhập tay thủ công.

+ Thiết kế giao diện GUI bằng C++ với thư viện như Qt hoặc dùng web interface.

+ Cho phép nhập dữ liệu từ bàn phím ngoài việc chỉ đọc file.

Mục tiêu phần vòng lặp:

Duyệt từng sinh viên → tìm xem sinh viên đó đã đăng ký môn nào → tính tổng số tín chỉ mà sinh viên ấy đã đăng ký.

B1: Lặp qua từng sinh viên trong danh sách students

B2: khởi tạo biến đếm totalCredits:tính tổng tín chỉ mà sinh viên này và vector registeredSubjects lưu danh sách môn sinh viên đã đăng kí

B3: Lặp qua từng bản ghi trong regists, kiểm tra xem bản đăng kí có phải của sinh viên hiện tại không nếu đúng là sinh viên lưu lại danh sách mã môn học mà bạn ấy đăng kí

B4: Lặp qua danh sách registeredSubjects nếu thấy môn học trùng mã sẽ thêm số tín chỉ của môn đó cộng thêm vào totalCredits

Từ totalCredits chúng ta có thể tính tổng số tiền mà sinh viên đó phải nộp vì tiền của mỗi tín chỉ là cố định