TRƯỜNG ĐẠI HỌC GIAO THÔNG VẬN TẢI

KHOA CÔNG NGHỆ THÔNG TIN

-------------o0o---------------



**Bài tập lớn môn học**

**CẤU TRÚC DỮ LIỆU VÀ GIẢI THUẬT**

**Đề tài số 26**

Giảng viên hướng dẫn : TS. Hoàng Văn Thông

Sinh viên thực hiện : Ngô Việt Đức (212641620)

Lớp : CNTT-VA 1

**Hà Nội, tháng 11 năm 2022**

Mục lục

[I. Đề bài 4](#_Toc119617073)

[II. Phân tích bài toán 5](#_Toc119617074)

[2.1 Yêu cầu bài toán 5](#_Toc119617075)

[2.2 Các lớp cần thiết 5](#_Toc119617076)

[2.3 Các thuộc tính 5](#_Toc119617077)

[a, Class vector\_reverse\_iterator. 5](#_Toc119617078)

[b, Class Vector. 5](#_Toc119617079)

[c, Claass Question. 5](#_Toc119617080)

[2.4 Các phương thức và chức năng 5](#_Toc119617081)

[a, Class vector\_reverse\_iterator. 5](#_Toc119617082)

[b, Class Vector. 6](#_Toc119617083)

[c, Class Question. 6](#_Toc119617084)

[III. Cài đặt các lớp và hàm main bằng C++ 7](#_Toc119617085)

[3.1 Class vector\_reverse\_iterator 7](#_Toc119617086)

[3.1.1 Hàm tạo 7](#_Toc119617087)

[3.1.2 Các phương thức truy cập 7](#_Toc119617088)

[3.2 Class Vector 7](#_Toc119617089)

[3.2.1 Hàm tạo, hủy 7](#_Toc119617090)

[3.2.2 Các phương thức truy cập 7](#_Toc119617091)

[3.2.3 Hàm lặp xuôi, ngược 8](#_Toc119617092)

[3.3 Class Question 8](#_Toc119617093)

[3.3.1 Hàm tạo, hủy 8](#_Toc119617094)

[3.3.2 Các phương thức truy cập 8](#_Toc119617095)

[3.4 Các hàm và thủ tục 9](#_Toc119617096)

[3.4.1 beginQuizz 9](#_Toc119617097)

[3.4.2 result 9](#_Toc119617098)

[3.4.3 crVec 9](#_Toc119617099)

[3.4.4 shVec 9](#_Toc119617100)

[3.5 Thiết lập hàm main 9](#_Toc119617101)

[IV. Phân tích thời gian chạy của từng phương thức trong lớp 11](#_Toc119617102)

[4.1 Class vector\_resverse\_iterator 11](#_Toc119617103)

[4.2 Class Vector 11](#_Toc119617104)

[4.3 Class Question 11](#_Toc119617105)

[4.4 Các hàm và thủ tục. 11](#_Toc119617106)

[V. Danh sách tài liệu tham khảo 12](#_Toc119617107)

**Lời mở đầu**

Cấu trúc dữ liệu và thuật toán là học phần rất quan trọng đối với người lập trình. Môn học này được xem như nền tảng của lập trình máy tính. Nó là cơ sở vững chắc để giải quyết một số bài toán, đồng thời cung cấp cho chúng ta hiểu biết về các giải thuật tác động lên dữ liệu, cũng như cách tổ chức dữ liệu để giải quyết các bài toán sao cho hiệu quả và tối ưu.

Sau khi học phần lí thuyết và thực hành, em đã nghiên cứu và thực hiện đề tài này theo sự hướng dẫn của TS. Hoàng Văn Thông, như là một cách để cũng cố và mở rộng kiến thức. Thông qua quá trình thực hiện đề tài, em đã nắm bắt được những kỹ thuật quan trọng của việc xây dựng cấu trúc dữ liệu và cách xây dựng một thuật toán sau cho tối ưu nhất.

Đề tài số 26 về vector và class đề thi mà em nghiên cứu và trình bày trong báo cáo sau đây.

Mong được sự góp ý và giúp đỡ từ thầy để đề tài này được hoàn thiện hơn.

# I. Đề bài

1. Cài đặt cấu trúc dữ liệu trừu tượng vector có bộ lặp xuôi và ngược

2. Cài đặt class câu hỏi trắc nghiệm mỗi câu hỏi gồm câu hỏi dạng văn bản và 4 phương án trong đó có 1 phương án đúng

3. Cài đặt class thi dùng cấu trúc vector đã xây dựng để đọc đữ liệu từ câu hỏi từ file văn bản lưu vào vector. Cho phép tạo ra đề thi gồm k câu hỏi ngẫu nhiên trong vector các câu hỏi, khi hiển thị sáo trộn ví trí các phương án trả lời cho phép người thi và tính điểm

# II. Phân tích bài toán

## 2.1 Yêu cầu bài toán

\* Cài đặt class cấu trúc dữ liệu trừu tượng, có bộ lặp ngược và xuôi.

\* Cài đặt class trắc nghiệm gồm câu hỏi với 4 đáp án và có 1 đáp án đúng.

\* Từ class vector nhập thông tin từ file văn bản vào vector xây dựng class đề thi. Đề thi gồm n vector câu hỏi và k câu hỏi lấy ngẫu nhiên từ n câu. Hiển thị đáp án ngẫu nhiên. Cho phép người thi và tính điểm.

## 2.2 Các lớp cần thiết

Class **vector\_reverse\_iterator**

Class **Vector**

Class **Question**

## 2.3 Các thuộc tính

### a, Class vector\_reverse\_iterator.

Kiểu hàm mẫu T: \*curr.

### b, Class Vector.

Kiểu nguyên: cap, num.

Kiểu hàm mẫu T: \*buff.

### c, Claass Question.

Kiểu chuỗi ký tự: Question\_Text, correct\_answer.

Kiểu mảng chuỗi ký tự: answer.

Kiểu nguyên: Question\_Score.

## 2.4 Các phương thức và chức năng

### a, Class vector\_reverse\_iterator.

Class để tạo bộ lặp ngược theo yêu cầu bài toán

vector\_reverse\_iterator: hàm tạo để tùy theo những kiểu dữ liệu đầu vào khác nhau mà có thể sử dụng.

### b, Class Vector.

Thiết lập các dữ liệu trừu tượng cho lớp vector.

- **capacity**: dùng để lấy độ chứa bộ nhớ khả dụng của ô chứa.

- **size**: dùng lấy độ dài hiện tại của vector.

- **empty**: kiểm tra xem vector có rỗng hay không.

- **pop\_back**: xóa một phần tử cuối trong vector.

- **extend**: mở rộng độ dài vector.

- **push\_back**: chèn vào cuối vector một giá trị.

- **insert**: chèn thêm vào một vị trí một giá trị.

- **iterator begin**, **iterator end**, **reverse\_iterator rbegin**, **reverse\_iterator rend**.

### c, Class Question.

Thiết lập class câu hỏi theo yêu cầu bài đặt ra.

**Question()**: tạo ra câu hỏi.

**setValues**: cài đặt giá trị ban đầu cho gồm: một câu hỏi, bốn đáp án, một đáp án đúng, số điểm cho câu hỏi cho class.

**askQuestion**: đầu vào là các giá trị đề bài cho, tính điểm, xáo trộn đáp án ngẫu nhiên, xuất ra các thông báo cần thiết để điều hướng và kết thúc một câu hỏi.

**getQuestion\_Text**: lấy một chuỗi câu hỏi từ class.

**getAnswers**: Lấy từng chuỗi một các đáp án đã lưu trong class.

**getCorrect\_answer**: Lấy một ký tự là đáp án đúng trong class.

# III. Cài đặt các lớp và hàm main bằng C++

## 3.1 Class vector\_reverse\_iterator

### 3.1.1 Hàm tạo

**vector\_reverse\_iterator(T \*c=0)** trả về giá trị hiện tại.

**vector\_reverse\_iterator<T> &operator=(vector\_reverse\_iterator<T> it)** trả về giá trị hiện tại cho it.

**vector\_reverse\_iterator<T> operator++()** trả về vị trí giảm dần

**vector\_reverse\_iterator<T> operator++(int)** trả về trả về giá trị giảm dần.

### 3.1.2 Các phương thức truy cập

**T &operator\*()** trả về giá trị hiện tại

**bool operator!=(vector\_reverse\_iterator<T> t)** trả về kết quả sai

## 3.2 Class Vector

### 3.2.1 Hàm tạo, hủy

**Vector()** tạo ra hàm không truyền tham số, khởi tạo ban đầu cho các thuộc tính bằng 0.

**Vector(int k, T x)** hàm truyền tham số, kiểu int là độ dài cũng như độ chứa tối đa của vector, hàm mẫu kiểu T để khởi tạo vector với độ dài k và độ chứa k,

dùng loop for(i) cài đặt giá trị cho mỗi phần tử của vector.

**~Vector()** hàm hủy xóa đi toàn bộ phần tử của Vector

**Vector &operator=(Vector<T> V)** đặt num, cap của Vector hiện tại cho V. Nếu (cap) thỏa mãn thì đặt các phần tử Vector hiện tại bằng các phần tử của V bằng loop for(i). Ngược lại thì các phần tử bằng 0.

### 3.2.2 Các phương thức truy cập

**Int capacity()** trả về một giá trị cap là độ chứa của Vector.

**Int size()** trả về một giá trị num là độ dài Vector.

**Bool emty()** trả về giá trị logic.

**Int pop\_back()** xóa phần tử cuối khi (num>0) thì giảm num đi 1 đớn vị.

**Int extend(int new\_cap)** nếu độ dài (new\_cap<cap) thì không làm gì, đặt cap=new\_cap và một biến động kiểu T \*temp tạo độ lưu trữ độ dài cap. Loop for(i) cho đến phần tử cuối của Vector và đặt mỗi phần tử của temp bằng Vector đã có và xóa vector ban đầu.

**Void push\_back(T x)** nếu (num=cap) thì gọi hàm mở rộng Vector và thêm phần tử x được truyền vào hàm.

**Insert(int k, T x)** nếu (cap=num) gọi hàm mở rộng Vector, loop for(i) cho đến phần tử thứ k thì tách ra một phần tử và đặt vị trí đó bằng x rồi ghi lại từ vị trí sau k là các phần tử còn lại.

**T &back()** trả về vị trí phần tử cuối.

**T &operator[](int k)** trả về phần tử thứ k.

### 3.2.3 Hàm lặp xuôi, ngược

**iterator begin()** trả về phần tử đầu.

**iterator end()** trả về phần tử cuối.

**reverse\_iterator rbegin()** trả về phần tử cuối.

**reverse\_iterator rend()** trả về phần tử đầu.

## 3.3 Class Question

### 3.3.1 Hàm tạo, hủy

**Question()** tạo ra câu hỏi.

### 3.3.2 Các phương thức truy cập

**void setValues(string, string, string, string, string, string, int)** truyền vào là câu hỏi, 4 câu trả lời, đáp án đúng, số điểm cho câu đó và đặt các giá trị này cho các thuộc tính class Question\_Text, answer[], correct\_answer, Question\_Score.

**void askQuestion()** ghi câu hỏi bằng phương thức **getQuestion\_Text()**. Đặt phạm vi random cho câu hỏi bằng cách sử dụng: **srand((unsigned int)time(NULL))**, **map<int, bool>** vis; sẽ tạo lập ngẫu nhiên không trùng lặp mỗi câu hỏi và không lặp lại câu hỏi đã hỏi. Loop for(i) cho đến 4 trong đó chạy lệnh tạo ra các số ngẫu nhiên sử dụng **srand** và **rand** từ 0-3 **r=rand()%4** trong loop while với điều kiện kết thúc khi chưa chạm đến phần tử cuối (vis.find(r)!=vis.end()).

Yêu cầu người thi nhập đáp án guess. Nếu (guess=**getCorrect\_answer()**) total+= Question\_Score. Ngược lại không trả về gì và tiếp tục chương trình.

**string getQuestion\_Text()** trả về câu hỏi.

**string getAnswers(int)** trả về bốn đáp án.

**string getCorrect\_answer()** trả về đáp án đúng.

## 3.4 Các hàm và thủ tục

### 3.4.1 beginQuizz

Đưa ra các câu lệnh điều hướng chương trình khi bắt đầu. Yêu cầu người thì trả lời yes (respond) để tiếp tục, hoặc sẽ kết thúc chương trình ngay.

### 3.4.2 result

Đưa ra các câu lệnh điều hướng chướng trình khi kết thúc. Và hiển thị ra màn hình số điểm đạt được(total) . Nếu lớn hơn mốc điểm nhất định được đặt ra thì vượt qua, hoặc không vượt qua.

### 3.4.3 crVec

**crVec(Vector<Question>& newVec, int n)** nhập thông tin từ tệp vào chương trình rồi đẩy vào Vector newVec.

### 3.4.4 shVec

**shVec(Vector<Question>& newVec, int n)** lấy ngẫu nhiên các **Question()** rồi thực hiện **askQuestion().**

## 3.5 Thiết lập hàm main

Cho chạy các thủ tục theo thứ tự:

1 beginQuizz() -> In ra màn hình câu lệnh nhập số câu hỏi muốn thi nhỏ hơn số câu hỏi đã lưu trong vector -> crVec() -> shVec() ->Tính điểm, sinh các câu hỏi ngẫu nhiên và đáp án ngẫu nhiên -> result().

# IV. Phân tích thời gian chạy của từng phương thức trong lớp

### 4.1 Class vector\_resverse\_iterator

Toàn bộ O(n) do chỉ gán và giảm tham số cơ bản.

### 4.2 Class Vector

Toàn bộ O(n) do chỉ gán.

### 4.3 Class Question

Toàn bộ O(n)

### 4.4 Các hàm và thủ tục.

**beginQuizz** O(1) do chỉ hiện thị ra màn hình và rẽ nhánh đơn giản .

**result** O(1) do chỉ hiện thị ra màn hình và rẽ nhánh đơn giản.

**crVec** O(n) một loop để nhập thông tin.

**shVec** O(n) một loop xuất thông tin

# V. Danh sách tài liệu tham khảo

https://drive.google.com/drive/u/0/folders/1\_j8xZmCPiGKqbOP\_5w8\_9GDI95uy42cI

https://drive.google.com/drive/u/0/folders/1NhJZ51tv4HPxthBsCo2Y4g3u23CPAFIg