

ĐẠI HỌC HUẾ  
Trường Đại học Kinh tế  
Khoa Hệ thống thông tin Kinh tế

**Cơ sở lập trình  
Bài tập**

*Hạn cuối nộp bài : 24h00 thứ ba, ngày 08 tháng 12 năm 2015*

*Nộp bằng cách gửi email về địa chỉ [lvman@hce.edu.vn](mailto:lvman@hce.edu.vn)*

*Hãy đọc kỹ **Quy định nộp báo cáo bài tập** trước khi làm bài*

## Giới thiệu

Trong bài tập này, các bạn sẽ vận dụng tất cả các khái niệm, kiến thức đã học trong môn Cơ sở lập trình và sử dụng ngôn ngữ C# để thiết kế, xây dựng một chương trình hoàn chỉnh cho phép quản lý các chương trình đào tạo.

## Quản lý chương trình đào tạo

Khoa Hệ thống Thông tin Kinh tế - Trường Đại học Kinh tế Huế có nhu cầu mua một phần mềm để quản lý các môn học trong chương trình đào tạo của chuyên ngành Tin học Kinh tế. Chương trình đào tạo chuyên ngành Tin học Kinh tế được thiết kế gồm 63 môn học, trong đó có 34 môn bắt buộc và 29 môn tự chọn. Sinh viên chuyên ngành Tin học Kinh tế phải tích lũy tối thiểu 122 tín chỉ (94 tín chỉ của các môn học bắt buộc và 28 tín chỉ của các môn học tự chọn).

Phần mềm này cho phép thêm, sửa, xóa các môn học trong chương trình đào tạo. Mỗi môn học có các thông tin sau được quản lý: mã học phần, tên học phần, số tín chỉ, điều kiện tiên quyết, môn tự chọn hay bắt buộc, học kỳ dự kiến.

Phần mềm còn phải có các chức năng sau:

- Xem thông tin của chương trình đào tạo chuyên ngành Tin học Kinh tế, gồm các thông tin sau: tên chương trình, số tín chỉ cần tích lũy để tốt nghiệp, số môn học bắt buộc, tổng số tín chỉ các môn bắt buộc, số môn học tự chọn và tổng tín chỉ của các môn tự chọn.
- Xem thông tin một môn học trong chương trình đào tạo
- In danh sách các môn học có thể đăng ký học tại một học kỳ nào đó
- In danh sách các môn học tiên quyết của một môn học nào đó

## Yêu cầu của bài tập

Hãy sử dụng tất cả các kiến thức về ngôn ngữ lập trình C# đã được học để lập trình tạo ra một ứng dụng quản lý đăng ký mượn phòng thực hành máy tính. Chương trình cần có các chức năng sau :

1. Có cấu trúc dữ liệu để lưu trữ danh sách các môn học trong chương trình đào tạo
2. Thêm một môn học vào chương trình đào tạo
3. Mỗi môn học có một mã học phần riêng biệt, phần mềm phải kiểm tra khi người sử dụng nhập vào để tránh bị trùng với mã học phần của môn học khác.
4. Tìm kiếm một môn học theo mã học phần
5. In danh sách các môn học có một phần tên trùng với từ khoá mà người sử dụng nhập vào khi tìm kiếm môn học theo tên học phần
6. Sửa thông tin môn học, không cho phép sửa mã môn học
7. Xoá một môn học. Khi xoá một môn học thì phải xoá môn đó ra khỏi điều kiện tiên quyết của các môn học khác
8. Xem thông tin một môn học trong chương trình đào tạo, có kèm danh sách các môn học tiên quyết của môn học đó
9. In danh sách các môn học có thể đăng ký học tại một học kỳ nào đó
10. Xem thông tin của chương trình đào tạo
11. (Tuỳ chọn - cộng điểm) Hỗ trợ đối số dòng lệnh
12. (Tuỳ chọn - cộng điểm) Lưu danh sách các môn học và thông tin của chương trình đào tạo vào file
13. (Tuỳ chọn - cộng điểm) Đọc danh sách các môn học và thông tin của chương trình đào tạo từ file
14. (Tuỳ chọn - cộng điểm) Dữ liệu đầu vào được kiểm tra và yêu cầu nhập lại khi dữ liệu không hợp lý

*Chú ý :*

1. Nên tổ chức chương trình dưới dạng hàm để dễ kiểm soát và sửa lỗi

2. Trong mỗi chức năng các bạn nên xây dựng thành các hàm chức năng nhỏ hơn. Như vậy, chương trình sẽ không bị rối và dễ gỡ lỗi. Đồng thời các chức năng khác có thể sử dụng lại các hàm đó.
3. Nên code từng phần một, sau mỗi phần phải kiểm tra kỹ càng đoạn code vừa được viết là hoạt động tốt rồi mới chuyển sang phần khác. *Ví dụ : sau khi code phần cho phép nhập dữ liệu vào, thì nên viết đoạn code in dữ liệu ra để kiểm soát việc nhập dữ liệu vào là đúng*
4. Khi kiểm tra thì nên kiểm tra với lượng ít số liệu để dễ kiểm soát chương trình. Sau khi chương trình đã đúng với số ít dữ liệu đó thì mới chạy chương trình với nhiều dữ liệu hơn

## Test

Các bạn dùng bộ test sau đây để kiểm tra chương trình và chụp ảnh màn hình kết quả để dán vào báo cáo.

1. Nhập dữ liệu các môn học sau : *Đây chỉ là dữ liệu test, nên có thể sai khác so với thực tế*

Mã MH	Tên môn học	Số TC	ĐKTQ	Loại MH	Học kỳ
DHKH1022	Nguyên lý CB CN Mác Lênin 1	2		BatBuoc	1
DHKH1112	Tư tưởng HCM	2	DHKH1022	BatBuoc	3
DHKH1103	Đường lối CM Đảng CSVN	3	DHKH1112	BatBuoc	4
DHKH1032	Khoa học môi trường	2		TuChon	1
KTPT1012	Địa lý kinh tế	2		TuChon	1
DHNN1013	Tiếng Anh 1	3		BatBuoc	1
DHNN1022	Tiếng Anh 2	2	DHNN1013	BatBuoc	2
DHNN1032	Tiếng Anh 3	2	DHNN1022	BatBuoc	3
HTTT1012	Toán cao cấp 1	2		BatBuoc	1
HTTT1023	Toán cao cấp 2	3		BatBuoc	1
HTTT1103	Tin học đại cương	3		BatBuoc	1
HTTT1033	Lý thuyết XSTK	3	HTTT1012 HTTT1023	BatBuoc	2
KTPT2023	Kinh tế vi mô	3	HTTT1012	BatBuoc	2
KTPT2033	Kinh tế vĩ mô	3	HTTT1012	BatBuoc	2
KTTC2013	Nguyên lý kế toán	3	KTPT2023 KTPT2033	BatBuoc	3
HTTT4403	Nguyên lý Thống kê KT	3	HTTT1033 KTPT2023	BatBuoc	3

			KTPT2033		
QTKD3023	Quản trị học	3	KTPT2023 KTPT2033	BatBuoc	3
HTTT2053	Kinh tế lượng	3	KTPT2023 KTPT2033 HTTT1033	BatBuoc	3
HTTT2043	Toán rời rạc	3	HTTT1012	BatBuoc	2
HTTT2123	Kiến trúc máy tính và HDH	3	HTTT1103 HTTT2043	BatBuoc	3
HTTT3153	Cơ sở dữ liệu	3	HTTT1103 HTTT2043	BatBuoc	3
HTTT3183	Cơ sở lập trình	3	HTTT1103 HTTT2043	BatBuoc	3
QTKD2013	Marketing căn bản	3	KTPT2023 KTPT2033	TuChon	3
KTTC2223	Tài chính - tiền tệ	3	KTPT2033	TuChon	3
HTTT2133	Hệ thống thông tin quản lý	3	HTTT1103 QTKD3023	BatBuoc	4
HTTT4283	ERP	3	QTKD3023 HTTT2133	BatBuoc	6
HTTT3173	Mạng và truyền thông	3	HTTT2123	BatBuoc	4
HTTT4203	Lập trình UD trong quản lý	3	HTTT3183 HTTT2133	BatBuoc	5
HTTT4233	CSDL nâng cao	3	HTTT3153 HTTT3183	TuChon	4
HTTT4313	Kỹ thuật TMDT	3		TuChon	6

2. In chương trình đào tạo
3. Sửa số tín chỉ của môn Toán cao cấp 2 xuống còn 2 tín chỉ
4. Bỏ môn tiên quyết Toán rời rạc ra khỏi môn Cơ sở lập trình
5. Xoá môn Toán cao cấp 1
6. In thông tin môn Cơ sở lập trình
7. In danh sách các môn học có thể đăng ký trong học kỳ 4
8. In lại chương trình đào tạo

## Hướng dẫn lập trình

Những thuật toán, cấu trúc dữ liệu và cách xử lý được đưa ra trong các phần sau đây chỉ là **đề nghị**. Các bạn có thể đưa ra thuật toán, cấu trúc dữ liệu và cách xử lý riêng và hãy chỉ ra trong báo cáo của bạn. Thuật toán, cấu trúc dữ liệu và cách xử lý của bạn là tốt hơn sẽ được đánh giá cao hơn và ngược lại, điểm của bạn sẽ thấp nếu chương trình của bạn hỗ trợ ít chức năng hơn hoặc cách xử lý không tốt hơn.

Các phần dưới đây sẽ được trình bày tuần tự theo một quy trình mà nếu các bạn làm theo từng bước thì có thể hoàn thành tốt chương trình. Các bước sau sẽ là phát triển tiếp của bước trước, nên không làm bước trước thì sẽ khó hoàn thành được bước sau.

### Cấu trúc dữ liệu để lưu Chương trình đào tạo

Cấu trúc dữ liệu là một trong hai thành phần cốt lõi để tạo nên một chương trình máy tính và nó có tính quyết định đến việc lựa chọn thuật toán, cách xử lý trong chương trình. Do đó, khi phân tích thiết kế chương trình, ta nên lựa chọn cấu trúc dữ liệu tốt để việc xử lý sau này sẽ đơn giản và thuận tiện hơn.

Đầu tiên, chúng ta cần phân tích xem chương trình cần lưu những dữ liệu gì. Theo mô tả của bài toán, chúng ta cần lưu trữ thông tin các môn học của chương trình đào tạo Tin học Kinh tế. Mỗi môn học cần lưu trữ các thông tin : mã học phần, tên học phần, số tín chỉ, các môn học điều kiện tiên quyết, thông tin để phân biệt đây là môn tự chọn hay bắt buộc. Như vậy, cấu trúc dữ liệu để lưu danh sách các môn học trên phải thoả mãn hai đặc điểm :

1. cho phép lưu trữ một loạt các số liệu
2. mỗi thành phần trong loạt số liệu đó bao gồm :
  - một chuỗi ký tự cho mã môn học
  - một chuỗi ký tự để lưu tên môn học
  - một số nguyên để lưu số tín chỉ
  - danh sách các mã môn học của các môn học tiên quyết
  - một biến kiểu true/false để phân biệt là môn tự chọn hay bắt buộc (vd: true là bắt buộc, false là tự chọn), hoặc một biến kiểu liệt kê enum.
  - một số nguyên cho học kỳ dự kiến

Với đặc điểm đầu tiên, bạn có thể dùng mảng hoặc List để lưu trữ một loạt các số liệu. Mảng rõ ràng là đơn giản nhất, nhưng mảng lại không có tính linh động khi chúng ta cần mở rộng kích thước của mảng, List thì linh động và mạnh mẽ hơn. Dù là mảng kém linh động như vậy nhưng chúng ta vẫn có thể dùng. Nếu sử dụng mảng, bạn sẽ cấp phát số thành phần

cố định ban đầu cho mảng và khi đó, bạn chỉ có thể lưu trữ nhiều nhất bằng số lượng cố định đó.

Với đặc điểm thứ hai, chỉ một mảng một chiều hoặc một `List` sẽ không thể nào phù hợp. Do đó, ta có thể sử dụng :

- *Cách 1* : nhiều mảng một chiều hoặc nhiều `List`. Mỗi mảng/`List` có kiểu dữ liệu phù hợp để lưu trữ cho *một* thông tin của môn học. Trong đó, thành phần thứ *i* trong mỗi mảng/`List` đó tương ứng với môn học thứ *i*.

*Ví dụ* (dùng nhiều mảng một chiều): Chúng ta sẽ dùng bốn mảng một chiều : hai mảng kiểu `string` để lưu mã môn học và tên môn học, hai mảng kiểu `int` để lưu số tín chỉ và học kỳ dự kiến, một mảng kiểu `bool` để lưu loại môn học (tự chọn hay bắt buộc). Với danh sách các mã môn học tiên quyết, ta có thể sử dụng một mảng hai chiều kiểu `string`.

- *Cách 2* : một mảng một chiều hoặc một `List` mà mỗi phần tử là một kiểu cấu trúc (`struct`). Cấu trúc nên được định nghĩa tốt để lưu trữ các thông tin của một môn học.

Ví dụ, chúng ta có thể định nghĩa cấu trúc như sau :

```
1 struct MonHoc
2 {
3     public string ma; // ma hoc phan
4     public string tenMH; // ten mon hoc
5     public int tinchi; // so tin chi
6     public List<string> monhocTQ; // danh sach ma
       mon hoc tien quyet
7     public LoaiMonHoc loaiMH; // loai mon hoc
8     public int hocky; // hoc ky du kien
9 };
10
11 enum LoaiMonHoc {TuChon, BatBuoc}; // kieu liet ke
       the hien hai loai mon hoc
```

## Nhập dữ liệu môn học

Trong việc nhập dữ liệu, bạn có thể áp dụng theo hai kiểu sau :

1. *Biết trước* số lượng môn học sẽ được nhập

Tức là, bạn sẽ yêu cầu người sử dụng nhập vào số lượng môn học muốn nhập. Sau đó, dựa vào giá trị số lượng đó, bạn có thể dùng một vòng lặp để cho phép người sử dụng nhập vào dữ liệu từng môn học.

## 2. Không biết trước số lượng môn học sẽ được nhập

Tức là, số lượng môn học không được biết trước (chương trình không hỏi giá trị này), nên sau khi người sử dụng nhập xong dữ liệu của một môn học nào đó thì chương trình sẽ hỏi xem người sử dụng có muốn nhập tiếp hay không. Tùy vào câu trả lời của người sử dụng mà chương trình cho phép người sử dụng tiếp tục nhập hoặc thoát ra khỏi tiến trình nhập này.

Hoặc một cách khác, các bạn có thể dừng việc nhập khi người sử dụng nhập vào ký tự kết thúc file (end of file - Ctrl + Z).

Xem ví dụ vòng lặp `while` ở slide số 15, chủ đề 5 (Các cấu trúc điều khiển 2).

Ví dụ - thuật toán đoạn chương trình nhập dữ liệu theo dạng không biết trước số lượng môn học sẽ được nhập, cấu trúc dữ liệu theo cách 2 :

**Đầu vào :** biến kiểu `List < MonHoc >` đã được khởi tạo hoặc đã có dữ liệu, gọi là `listMonHoc`

**Đầu ra :** `listMonHoc` đã thêm vào các môn học mới

Khai báo một biến kiểu `MonHoc`, gọi là `monhoc`

Cho người sử dụng nhập vào mã môn học mới

Nếu mã môn học trùng với những môn học đã có

Thông báo trùng dữ liệu với môn học đã có

Chuyển đến phần Hỏi người sử dụng có muốn nhập tiếp môn học mới hay không

Ngược lại

Gán mã môn học cho thành phần `ma`

Cho người sử dụng nhập vào tên môn học và gán cho thành phần `tenMH`

Cho người sử dụng nhập vào số tín chỉ

Nếu số tín chỉ không phải là số, không nằm trong đoạn 1 - 10

Yêu cầu người dùng nhập lại

Ngược lại thì gán số tín chỉ cho thành phần `tinchi`

Cho người sử dụng nhập vào loại môn học

Nếu không là chuỗi "TuChon" hoặc "BatBuoc"

Yêu cầu nhập lại

Ngược lại thì gán cho thành phần `loaiMH`

Cho người sử dụng nhập vào học kỳ dự kiến

Nếu không phải số hoặc không nằm trong đoạn 1 - 4

Yêu cầu nhập lại

Ngược lại thì gán cho thành phần `hocky`

Cho người sử dụng nhập vào danh sách các mã môn học tiên quyết

Nếu có mã môn học chưa có trong hệ thống

Yêu cầu nhập lại

Ngược lại thì nhập vào thành phần *monhocTQ*

Gán biến *monhoc* vào *listMonHoc*

Hỏi người sử dụng có muốn nhập tiếp môn học mới hay không.

Nếu có, thì trở lại bước đầu tiên

Ngược lại, thoát ra khỏi thuật toán này

Cách tạo danh sách môn học như thuật toán trên sẽ cho ta một danh sách được sắp theo thứ tự nhập vào (Vì với mỗi đơn mới ta chỉ thêm vào cuối (sử dụng hàm **Add** của **List**) danh sách hiện có). Danh sách như thế này là không tốt cho các thao tác lọc dữ liệu hay thống kê. Bạn có thể sắp xếp danh sách này theo mã môn học, hoặc tên môn học hoặc theo học kỳ dự kiến.

Có hai cách để tạo ra danh sách được sắp xếp như vậy (ví dụ này sắp theo học kỳ dự kiến):

1. Sử dụng thuật toán sắp xếp để sắp xếp danh sách theo thứ tự mong muốn
2. Khi có một môn học mới, bạn không *thêm* vào mà bạn *chèn* nó vào vị trí thích hợp với thứ tự sắp xếp mong muốn  
Nếu bạn dùng cấu trúc kiểu **List** thì bạn dùng hàm **Insert** để chèn môn học mới vào danh sách. Còn bạn dùng mảng thì tham khảo câu 4, phần Mảng, bài thực hành 5.

*Những điều trình bày trên đây cũng phù hợp với cấu trúc dữ liệu mảng.*

*Sau đoạn chương trình nhập dữ liệu, bạn nên có đoạn chương trình cho phép in ra dữ liệu đã được nhập vào để kiểm tra lại dữ liệu đã nhập vào có đúng không.*

## Tìm kiếm một môn học theo mã môn học

Với cấu trúc dữ liệu theo cách 2, thuật toán tìm kiếm theo mã môn học như sau :

*Đầu vào* : mã môn học cần tìm

*Đầu ra* : *i* - vị trí của môn học tìm được trong danh sách  
, bằng -1 nếu không tìm được

Khởi gán  $i = 0$

Nếu  $i >$  số lượng phần tử trong danh sách

Thông báo là không có môn học cần tìm

Trả ra giá trị  $i = -1$

Ngược lại,

Nếu phần tử thứ *i* trong danh sách có biến thành phần *ma* bằng  
với mã môn học cần tìm

Kết thúc, giá trị của *i* là vị trí môn học cần tìm trong danh sách

Ngược lại, tăng giá trị của *i* lên một đơn vị

Quay lại bước 2 trong thuật toán này



## Tìm kiếm môn học theo một phần tên môn học

Với cấu trúc dữ liệu theo cách 2, thuật toán tìm kiếm các môn học theo một phần tên môn học như sau :

*Đầu vào* : chuỗi ký tự người sử dụng nhập vào để tìm

*Đầu ra* : danh sách các môn học có tên chứa chuỗi ký tự cần tìm

Duyệt qua danh sách các môn học

Kiểm tra tên của mỗi môn học có chứa chuỗi ký tự đầu vào hay không ?

Nếu có, hiển thị thông tin môn học đó

Ngược lại, duyệt tiếp môn học tiếp theo

## Sửa thông tin một môn học

Để sửa thông tin một môn học nào đó, ta có thể sử dụng thuật toán sau (cho cấu trúc dữ liệu theo cách 2) :

Yêu cầu người sử dụng nhập vào mã môn học của môn học muốn sửa

Sử dụng thuật toán *tìm kiếm môn học theo mã môn học* để lấy

vị trí của môn học trong danh sách

Nếu kết quả của thuật toán tìm kiếm là  $-1$  ( $i == -1$ ),

Thông báo không có môn học với mã muốn tìm, yêu cầu nhập lại nếu muốn

Nếu có nhập lại thì quay lại bước 1 trong thuật toán này

Nếu không thì thoát khỏi thuật toán

Ngược lại,

In thông tin hiện có của môn học ra màn hình

Yêu cầu người sử dụng nhập dữ liệu mới

Gán dữ liệu mới vào

## Xoá một môn học

Tương tự thuật toán sửa thông tin một môn học, thuật toán xoá một môn học như sau :

Yêu cầu người sử dụng nhập vào mã môn học của môn học muốn xoá

Sử dụng thuật toán *tìm kiếm môn học theo mã môn học* để lấy vị trí của môn học trong danh sách

Nếu kết quả của thuật toán tìm kiếm là  $-1$  ( $i == -1$ ),

Thông báo không có môn học cần tìm, yêu cầu nhập lại nếu muốn

Nếu có nhập lại thì quay lại bước 1 trong thuật toán này

Nếu không thì thoát khỏi thuật toán

Ngược lại,

In thông tin hiện có của môn học ra màn hình

Xác nhận yêu cầu xoá của người sử dụng

Nếu người sử dụng xác nhận xoá thì thực hiện xoá  
Nếu không, thì thoát khỏi thuật toán

## Xem thông tin môn học

Bạn có thể cho phép người dùng nhập vào mã môn học hoặc tên môn học. Sau đó, bạn duyệt qua danh sách các môn học để tìm ra đến môn học cần in thông tin. Nếu có thì in thông tin của môn học đó ra. Với danh sách các môn học tiên quyết, bạn nên in ra cả mã môn học và tên môn học.

## Xem danh sách môn học có thể đăng ký trong học kỳ

Bạn cho phép người dùng nhập vào học kỳ muốn xem. Sau đó, bạn duyệt qua danh sách các môn học. Môn học nào có thành phần *hocky* bằng với giá trị người dùng nhập vào thì in thông tin của môn học đó ra.

Chức năng này và chức năng ở mục Xem thông tin môn học đều thực hiện in thông tin môn học. Cho nên, bạn có thể viết một hàm riêng cho việc in thông tin môn học với đầu vào là mã môn học.

## Xem thông tin chương trình đào tạo

Trong các thông tin cần in ra trong chức năng này, bạn cần thống kê được bốn thông tin sau:

- số môn học bắt buộc
- tổng tín chỉ các môn bắt buộc
- số môn học tự chọn
- tổng tín chỉ các môn tự chọn

Bạn chỉ cần tạo ra bốn biến kiểu `int` để lưu trữ bốn thông tin trên. Bạn cho duyệt qua danh sách các môn học. Với mỗi môn học, bạn kiểm tra xem đó là môn tự chọn hay là bắt buộc. Nếu là bắt buộc, bạn tăng giá trị của biến lưu số môn học bắt buộc và cộng dồn số tín chỉ của môn học đó vào biến lưu tổng tín chỉ các môn bắt buộc. Nếu ngược lại thì làm tương tự cho hai biến lưu số môn học tự chọn và tổng tín chỉ các môn tự chọn.

## Menu chương trình

Trong các chương trình có nhiều chức năng, thì người lập trình thường hỗ trợ một menu chương trình. Người sử dụng chỉ việc chọn một số hoặc một chữ cái để thực hiện một chức năng nào đó. Khi một chức năng nào đó đã thực hiện xong thì menu đó lại được in ra lại và chương trình

hỏi người sử dụng tiếp tục muốn thực hiện chức năng nào. Chương trình chỉ kết thúc/dừng khi người sử dụng chọn số hoặc chữ cái tương đương chức năng dừng chương trình.

Ví dụ menu chương trình có thể như sau khi chương trình chạy :

CHUONG TRINH QUAN LY CHUONG TRINH DAO TAO

- 1 - Them mon hoc moi
- 2 - Sua mon hoc
- 3 - Xoa mon hoc
- 4 - Danh sach mon hoc theo ten mon hoc
- 5 - Xem thong tin mon hoc
- 6 - Xem danh sach mon hoc co the dang ky trong hoc ky
- 7 - Xem thong tin chuong trinh dao tao
- 8 - Doc du lieu tu file
- 9 - Ghi du lieu ra file
- 10 - Thoat

Ban chon so may : \_

Để hỗ trợ tính năng này, người ta thường dùng vòng lặp `while`. Và khi người sử dụng nhập số là 10 (theo ví dụ trên) thì mới ngắt vòng lặp `while`. Xem slide 23, chủ đề 5 - Các cấu trúc điều khiển (phần 2) để biết cách tạo một menu như trên.

## Xử lý đối số dòng lệnh

Xem slide 37-40, chủ đề 6.

Với chương trình này, các bạn có thể hỗ trợ 4 dạng đối số dòng lệnh sau :

```
1 qlctdt.exe
2 qlctdt.exe <num>
3 qlctdt.exe <path>
4 qlctdt.exe -h
```

Với `qlctdt.exe` là tên chương trình, `qlctdt` là viết tắt của **Quản lý chương trình đào tạo**.

Dạng thứ nhất chỉ gọi chương trình mà không đưa vào đối số nào cả. Như vậy, chương trình sẽ in ra menu chương trình và chờ người sử dụng lựa chọn chức năng tiếp theo để thực hiện.

Dạng thứ hai gọi chương trình kèm với một số (`<num>`). Chương trình sẽ thực hiện chức năng nhập `<num>` môn học mới. Đây là cách nhập dữ liệu khi biết trước số lượng môn học được nhập.

Dạng thứ ba là gọi chương trình kèm theo đường dẫn (`<path>`) đến tập tin văn bản chứa dữ liệu. (Đọc thêm phần Nhập dữ liệu từ file văn bản bên dưới)

Dạng thứ tư kèm theo đối số `-h` để yêu cầu in ra hướng dẫn sử dụng của chương trình.

Thuật toán của phần xử lý đối số dòng lệnh như sau :

Kiểm tra số lượng đối số dòng lệnh

Nếu lớn hơn 3 thì thông báo lỗi và in ra hướng dẫn sử dụng chương trình

Nếu không có đối số nào thì chạy chương trình bình thường

Nếu có 1 đối số

Đối số đó là chuỗi -h (so sánh chuỗi ký tự trong đối số với chuỗi “-h”) thì in ra hướng dẫn sử dụng

Nếu thuộc dạng 2 (thử đổi chuỗi thành số) thì cho nhập dữ liệu theo dạng biết trước số môn học sẽ được nhập

Nếu thuộc dạng 3 thì chuyển đến nhập dữ liệu từ file văn bản

**Chú ý :** Nếu bạn không làm yêu cầu 12 và 13 thì có thể bỏ qua việc xử lý dạng thứ 3 của đối số dòng lệnh

## Nhập dữ liệu từ file văn bản

Nội dung của tập tin chứa dữ liệu các môn học có thể được quy định như sau :

- Mỗi dòng chứa dữ liệu cho một môn học, bao gồm 6 thành phần phân tách với nhau bằng dấu phẩy (,)
  - thành phần thứ nhất là mã môn học
  - thành phần thứ hai là tên môn học
  - thành phần thứ ba là số tín chỉ
  - thành phần thứ tư là danh sách các mã môn học tiên quyết, được phân tách với nhau bằng dấu gạch ngang (–)
  - thành phần thứ năm là loại môn học (tự chọn hay bắt buộc)
  - thành phần thứ sáu là học kỳ dự kiến

Bởi vì tên người mượn có thể có khoảng trống nên phải dùng dấu phẩy để phân tách giữa các thành phần của mỗi dòng.

Ví dụ sau đây là một phần nội dung của tập tin test.txt :

```
1 DHKH1022,Nguyen ly co ban CN Mac Lenin 1,2,,BatBuoc,1
2 DHKH1112,Tu tuong HCM,2,DHKH1022,BatBuoc,3
3 DHKH1103,Duong loi CM Dang CSVN,3,DHKH1112,BatBuoc,4
4 DHKH1032,Khoa hoc moi truong,2,,TuChon,1
5 KTPT1012,Dia ly kinh te,2,,TuChon,1
6 DHNN1013,Tieng Anh 1,3,,BatBuoc,1
7 DHNN1022,Tieng Anh 2,2,DHNN1013,BatBuoc,2
8 DHNN1032,Tieng Anh 3,2,DHNN1022,BatBuoc,3
```

Thuật toán đọc dữ liệu từ file cho cấu trúc dữ liệu cách 2 như sau :

**Đầu vào** : biến kiểu *List < MonHoc >* đã được khởi tạo hoặc đã có dữ liệu, gọi là *listMH*

**Đầu ra** : *listMH* đã thêm vào các môn học mới được đọc từ file

Đọc dòng đầu tiên

Nếu đọc được dữ liệu (1 dòng văn bản)

Khởi tạo một biến kiểu *MonHoc*, gọi là *monhoc*

Dùng hàm Split tách các thành phần trong chuỗi đọc được gán vào biến *listTP*

Nếu *listTP* không đủ 6 thành phần

Thông báo file dữ liệu bị lỗi, thoát khỏi thuật toán

Ngược lại,

Gán thành phần đầu tiên trong *listTP* vào thành phần *ma* của biến *monhoc*

Gán thành phần thứ hai trong *listTP* vào thành phần *tenMH* của biến *monhoc*

Gán thành phần thứ ba trong *listTP* vào thành phần *tinchi* của biến *monhoc*

Dùng hàm Split để tách các thành phần trong thành phần thứ tư của *listTP*, rồi gán vào thành phần *monhocTQ* của biến *monhoc*

Gán thành phần thứ năm và sáu của *listTP* vào hai thành phần *loaiMH* và *hocky* của biến *monhoc*

Đọc dòng tiếp theo trong file, quay lại bước 2 của thuật toán này

## Xuất dữ liệu ra file văn bản

Thuật toán xuất dữ liệu ra file văn bản như sau :

**Đầu vào** : biến kiểu *List < MonHoc >* đã được khởi tạo hoặc đã có dữ liệu, gọi là *listMH*

**Đầu ra** : file văn bản chứa dữ liệu các đơn mượn phòng

Duyệt qua từng môn học trong *listMH*, với mỗi môn học

Sử dụng cú pháp token giữ chỗ để tạo ra một chuỗi ký tự chứa 5 thành phần (mã môn học, tên môn học, số tín chỉ, danh sách các môn học tiên quyết, loại môn học và học kỳ dự kiến) theo cú pháp của file dữ liệu đã quy định, các thành phần phân tách nhau bởi dấu phẩy.

Trong đó, các mã môn học trong chuỗi danh sách các môn học tiên quyết phải cách nhau bằng dấu gạch ngang.

Ghi chuỗi ký tự được tạo ra vào file.

*Chúc các em làm bài tốt !*