

# SỞ BÁO CÁO THỰC TẬP CƠ

Giáo viên hướng dẫn:

Nhóm thực hiện:

PGS-TS Mai Văn Trinh

Hoàng Thị Thúy (Nhóm trưởng)

Nguyễn Bá Hùng

Phạm Hồng Phúc

Nguyễn Việt Quân

**TRƯỜNG ĐẠI HỌC VINH**  
**Khoa Công nghệ thông tin**

**ĐỀ TÀI THỰC TẬP CƠ SỞ**

**Mã đề: 043**

**Câu 1: Các khái niệm cơ bản về mạng máy tính**

Khái niệm mạng máy tính  
Tìm hiểu dịch vụ web  
Tìm hiểu dịch vụ thư điện tử  
Kết nối máy tính với internet

**Câu 2: Các phương pháp tìm kiếm trong** (tìm kiếm tuần tự, tìm kiếm nhị phân, cây nhị phân tìm kiếm)

Yêu cầu:

Trình bày thuật toán  
Ví dụ minh họa  
Cài đặt thuật toán trên 1 ngôn ngữ nào đó (Pascal, C, C++ ... )

**Câu 3: Tìm hiểu và sử dụng Outlook Express**

**Câu 4: Từ điển**

Cho tệp văn bản trong đó có chứa các từ, các dấu phân cách từ: dấu cách, dấu phẩy, dấu chấm, dấu chấm phẩy, dấu hai chấm, dấu chấm than, dấu hỏi. Mọi từ bắt đầu bằng chữ cái 'A'..'Z' (không phân biệt chữ hoa và chữ thường).

1. Viết thủ tục đọc các từ trong tệp văn bản và lưu trữ vào mảng các danh sách liên kết:

```
Type DanhSach=^PhanTu;  
PhanTu=Record  
Tu:String[15];  
Tiep:DanhSach;  
End;  
Var
```

TuDien: array['A'..'Z'] of DanhSach;

Mỗi danh sách liên kết trong từ điển xếp theo thứ tự tăng và các từ khóa là khác nhau.

2. Viết thủ tục hiển thị các từ ra màn hình theo thứ tự tăng;
3. Viết thủ tục bổ sung từ mới bằng cách đọc từ bàn phím và tìm nó trong từ điển.
  - Nếu thấy, hiển thị thông báo “Đã có”.
  - Nếu không, chèn vào vị trí thích hợp.
4. Viết Menu để thực hiện các chức năng trên.

## Phân công công việc

### Nhiệm vụ:

**Trưởng nhóm:** Chịu trách nhiệm phân công công việc, tổng hợp báo cáo của các thành viên trong nhóm.

**Các thành viên:** Tìm hiểu khái quát về các nội dung Thực tập. Nghiên cứu, phân tích và thực hiện các yêu cầu cụ thể các vấn đề đã được phân công, viết báo cáo về các nội dung đã thực hiện.

### Phân công công việc:

Câu 1: Các khái niệm cơ bản về mạng máy tính

Người thực hiện: Phạm Hồng Phúc

Nội dung: Tìm hiểu các khái niệm cơ bản về mạng máy tính

- o Khái niệm mạng máy tính
- o Tìm hiểu dịch vụ web
- o Tìm hiểu dịch vụ thư điện tử
- o Kết nối máy tính với internet

Câu 2: Các phương pháp tìm kiếm trong (tìm kiếm tuần tự, tìm kiếm nhị phân, cây nhị phân tìm kiếm)

Người thực hiện: Nguyễn Bá Hùng

Nội dung: Tìm hiểu các phương pháp tìm kiếm trong

- o Trình bày thuật toán
- o Ví dụ minh họa
- o Cài đặt thuật toán trên 1 ngôn ngữ nào đó (Pascal, C, C++ ...)

Câu 3: Tìm hiểu và sử dụng Outlook Express

Người thực hiện: Hoàng Thị Thúy

Nội dung: Tìm hiểu, cài đặt, thiết lập và sử dụng Outlook Express

Câu 4: Từ điển

Người thực hiện: Nguyễn Viết Quân

Nội dung: Tìm hiểu giải thuật, phương pháp và giải bài toán Từ điển bằng Pascal.

## Nội dung báo cáo

### ĐỀ TÀI THỰC TẬP CƠ SỞ

Mã đề: 043

#### Câu 1: Các khái niệm cơ bản về mạng máy tính

Khái niệm mạng máy tính

Tìm hiểu dịch vụ web

Tìm hiểu dịch vụ thư điện tử

Kết nối máy tính với internet

---

### Bài làm

#### 1.1 Khái niệm mạng máy tính

##### 1.1.1 Định nghĩa mạng máy tính

Mạng máy tính là một tập hợp các máy tính đơn lẻ được kết nối với nhau bằng phương tiện truyền vật lý (Transmission Medium) và theo một kiến trúc mạng xác định (Network Architecture).

Kiến trúc mạng bao gồm : cấu trúc mạng (Topology) và giao thức mạng (Protocol).

Topology là cấu trúc hình học của các thực thể mạng. Protocol là tập các quy tắc chuẩn theo đó các thực thể hoạt động trong truyền thông phải tuân theo.

##### 1.1.2. Lợi ích của mạng máy tính

Chia sẻ và dùng chung các tài nguyên

Giảm số lượng các thiết bị vật lý trong mạng

Quản lý tập trung

Tính bảo mật và an toàn dữ liệu

##### 1.1.3. Kiến trúc mạng

###### 1.1.3.1. Cấu trúc mạng (Topology)

Topology là cấu trúc hình học không gian của mạng. Mỗi máy tính khi tham gia trong một mạng được xem như là một node phân biệt với nhau.

Topology thực chất là cách bố trí vật lý các node và các thức kết nối chúng lại với nhau. Có 2 kiểu cấu trúc mạng :

Kiểu Điểm – Điểm (Point to Point): Đường truyền nối từng cặp node lại với nhau theo một cấu trúc hình học xác định

Kiểu Quảng bá (Multi Point) : Tất cả các node cùng truy nhập chung trên một đường truyền vật lý. Một thông điệp gửi từ một node sẽ được tất cả các node còn lại trong mạng tiếp nhận. Các mạng có cấu trúc kiểu Quảng bá được chia thành 2 loại : Quảng bá tĩnh và Quảng bá động

#### 1.1.3.2. Giao thức mạng máy tính (Protocol)

Các thực thể trong mạng muốn trao đổi thông tin với nhau phải thông qua một số thủ tục, quy tắc được quy định trước. Nghĩa là phải “Nói chung một ngôn ngữ”. Tập các quy tắc hội thoại dùng trong mạng máy tính được gọi là Giao thức mạng (Protocol). Giao thức truyền thông trong mạng máy tính là bộ giao thức TCP/IP.

#### 1.1.4. Phương tiện truyền vật lý

Phương tiện truyền vật lý là vật truyền tải các tín hiệu điện tử giữa các thành phần mạng với nhau. Bao gồm các loại cáp và phương tiện vô tuyến.

##### 1.1.4.1. Các loại cáp mạng

###### a. Cáp đồng trục

Cáp đồng trục có hai đường dây dẫn và có chung một đường trục, một dây dẫn trung tâm ( thường là dây đồng cứng ), đường dây còn lại tạo thành đường ống bao xung quanh dây dẫn trung tâm ( dây dẫn này có thể là dây bện kim loại và có chức năng chống nhiễu nên được gọi là lớp bọc kim ). Giữa hai dây dẫn trên có một lớp cách ly và ngoài cùng là lớp vỏ Plastic để bảo vệ cáp. Cáp đồng trục sử dụng đầu nối chữ T và BNC

Hai loại cáp thường sử dụng:

Cáp đồng trục mỏng

Cáp đồng trục dày

###### b. Cáp xoắn đôi

Gồm hai đường dây dẫn xoắn lại với nhau nhằm mục đích làm giảm nhiễu điện từ. Là loại cáp rẻ và dễ cài đặt tuy nhiên chịu tác động nhiễu của môi trường.

Có hai loại cáp xoắn đôi :

Cáp có bọc kim loại STP ( Shield Twisted Pair )

Cáp không bọc kim loại UTP ( Unshield Twisted Pair )

c. Cáp quang

Cáp sợi quang bao gồm một dây dẫn trung tâm ( là một sợi hay một bó sợi thủy tinh có thể truyền dẫn tín hiệu quang ) được bọc một lớp vỏ bọc có tác dụng phản xạ các tín hiệu trở lại để giảm sự mất mát tín hiệu trên đường truyền. Ngoài cùng là lớp vỏ Plastic bảo vệ cáp. Cáp quang không truyền dẫn các tín hiệu điện mà chỉ truyền dẫn các tín hiệu quang do đó các tín hiệu dữ liệu phải được chuyển đổi thành các tín hiệu quang trước khi truyền trên cáp quang và khi nhận lại ở bên thu chúng sẽ được chuyển đổi trở lại thành tín hiệu điện.

1.1.4.2. Phương tiện vô tuyến

a. Radio

Quang phổ của điện từ nằm trong khoảng 10Khz đến 1Ghz. Có nhiều dải tần của Radio : Sóng ngắn ( Short Wave ), VHF ( Very Hight Frequency ) , UHF ( Ultra Hight Frequency ) , FM radio, Tivi....

Đặc tính truyền :

Tần số đơn, công suất thấp không hỗ trợ tốc độ dữ liệu các mạng cục bộ LAN yêu cầu

Tần số đơn, công suất cao dễ cài đặt , băng thông cao từ 1 đến 10 Mb/s. Suy hao chậm, khả năng nhiễu từ thấp, bảo mật kém.

b. Viba

Truyền thông Viba có hai dạng : Viba mặt đất và Viba vệ tinh. Viba mặt đất sử dụng các trạm thu và phát. Kỹ thuật truyền thông vệ tinh sử dụng các trạm thu mặt đất ( các đĩa vệ tinh ) và các vệ tinh. Thời gian truyền một tín hiệu độc lập với khoảng cách. Thời gian trễ lan truyền ( Proragation Delay ) từ vệ tinh đến các trạm trong vòng 1/3 trái đất khoảng 0,5 đến 5 giây.

1.1.5. Phân loại mạng máy tính

1.1.5.1. Mạng cục bộ LAN (Local Area Network )

Mạng cục bộ là mạng được thiết lập để liên kết các máy tính trong một khu vực như trong một tòa nhà , xí nghiệp, doanh nghiệp....

Có 2 loại mạng LAN khác nhau : mạng LAN nối dây ( Sử dụng các loại cáp mạng ) và mạng LAN không dây ( Sử dụng sóng cao tần hay tia hồng ngoại ).

Đặc trưng cơ bản của mạng cục bộ LAN:



### Quy mô của mạng nhỏ

Mạng cục bộ sử dụng công nghệ truyền dẫn thường là Quảng bá.

Cấu trúc mạng ( Topology ) đa dạng : Bao gồm mạng hình Bus, mạng hình sao ( Start ) , mạng hình vòng ( Ring ) và các loại mạng kết hợp

#### a. Mạng hình BUS

Trong mạng hình BUS các máy tính đều được nối vào một đường truyền chính (BUS). Khi một trạm truyền dữ liệu, tín hiệu được truyền trên cả hai chiều của đường truyền theo từng gói một, mỗi gói đều phải mang địa chỉ trạm đích. Gói dữ liệu sẽ được quảng bá đến tất cả các trạm trong mạng. Mỗi trạm đều nhận dữ liệu đi qua, kiểm tra nếu đúng với địa chỉ của mình thì nhận gói tin nếu không thì bỏ qua.

#### b. Mạng hình sao

Trong mạng hình sao, tất cả các trạm đều được nối vào một thiết bị trung tâm có nhiệm vụ nhận tín hiệu từ các trạm và chuyển tín hiệu đến trạm đích. Tùy theo yêu cầu truyền thông trong mạng, thiết bị trung tâm có thể là một bộ chuyển mạch ( Switch ) , bộ chọn đường ( Router ) hoặc là bộ phân kênh ( Hub ) có nhiều cổng ra và mỗi cổng nối với một máy tính.

#### c. Mạng hình vòng

Trong mạng hình vòng ( RING ) các máy tính được liên kết với nhau theo một vòng tròn. Tín hiệu được truyền trong vòng theo một chiều duy nhất. Mỗi gói dữ liệu đều mang địa chỉ trạm đích. Mạng hình vòng là một mạng kiểu Quảng bá nên khi một trạm nhận được gói dữ liệu nó sẽ kiểm tra địa chỉ có đúng với địa chỉ của nó hay không, nếu đúng thì nó nhận gói dữ liệu, nếu không thì nó sẽ phát lại cho trạm kế tiếp theo chiều đã quy định.

#### 1.1.5.2 Mạng đô thị MAN ( Metropolitan Area Network )

Mạng đô thị MAN là tập hợp của các mạng LAN và hoạt động theo kiểu quảng bá. Được gọi là mạng đô thị vì phạm vi của nó bao gồm toàn bộ một thành phố hay đô thị. Mạng cung cấp các dịch vụ thoại , phi thoại và truyền hình cáp.

#### 1.1.5.3 Mạng WAN ( Wide Area Network )

Trong mạng diện rộng WAN, các máy tính được nối vào một mạng con, trong đó có chứa các bộ chọn đường (Routers) và các đường truyền tải (Transmission lines). Các Router làm nhiệm vụ lưu và chuyển tiếp gói tin mà nó nhận được theo nguyên lý cơ bản sau : Các gói tin đến một Router sẽ

được lưu vào trong một hàng chờ, tiếp theo Router sẽ kiểm tra và so sánh các thông tin của gói tin với bảng chọn đường của Router để quyết định địa chỉ đích của gói tin và sẽ chuyển gói tin lên đường được chọn.

**Đặc trưng cơ bản của một mạng WAN**

Hoạt động trên phạm vi một quốc gia hoặc trên toàn cầu

Tốc độ truyền dữ liệu thấp hơn so với mạng cục bộ

Lỗi truyền cao

**Một số mạng WAN điển hình**

Mạng tích hợp dịch vụ số ISDN (Integrated Services Digital Network)

Mạng X25 và chuyển mạch khung Frame Relay

## **1.2 Tìm hiểu dịch vụ web**

### **1.2.1 Khái niệm**

Dịch vụ World Wide Web gọi tắt là Web là một mạng lưới không gian thông tin toàn cầu mà người dùng có thể truy nhập qua các máy tính nối với mạng Internet.

Các tài liệu trên Web được lưu trữ trong một hệ thống siêu văn bản (HTML) đặt tại các máy tính trên Internet. Người dùng phải sử dụng một chương trình gọi là trình duyệt Web (Web Browser) để có thể xem được siêu văn bản đó. Trình duyệt nhận thông tin về địa chỉ do người sử dụng nhập vào (tên miền) rồi gửi thông tin đó đến máy chủ (Web Server), máy chủ tiếp nhận thông tin, xử lý và gửi nội dung phản hồi về cho người dùng hiển thị trên máy tính. Các trình duyệt web phổ biến và mạnh mẽ nhất hiện nay là : Firefox, Internet Explorer, Chrome ...

### **1.2.2 Lợi ích của dịch vụ Web đối với đời sống con người**

Internet phát triển kết hợp với dịch vụ Web thật sự đã đem lại cho con người nguồn tài nguyên vô tận mà không phải tốn nhiều công sức cũng có thể khám phá được nó. Sau 20 năm phát triển, giờ đây với những công nghệ mới và tiên tiến hơn rất nhiều, dịch vụ Web đã trở nên gần gũi với người sử dụng và đáp ứng tối đa những yêu cầu trong cuộc sống, công việc cũng như trong mọi lĩnh vực của xã hội như : thương mại, giải trí, giáo dục, nghiên cứu khoa học... Những lợi ích của Website :

Tìm kiếm thông tin, tài liệu...

Giải trí, nghe nhạc, chơi game trực tuyến

Thương mại điện tử

Học trực tuyến....

Đối với dịch vụ Web, nó cho phép con người có thể sử dụng các ngôn ngữ lập trình để thao tác với các trang Web (còn gọi là Website). Thông qua các Website này mà người dùng có thể khai thác các dịch vụ, tiện ích mà Website đem lại.

Các ngôn ngữ lập trình Web phổ biến hiện nay là :

ASP, ASP.NET, C# (Nền tảng công nghệ Microsoft)

PHP (Mã nguồn mở)

JAVA (Nền tảng công nghệ của SUN)

AJAX (Web 2.0)

### 1.2.3 Sử dụng dịch vụ Web

#### a. Tìm kiếm thông tin trên Internet

Để sử dụng dịch vụ Web tìm kiếm thông tin về các lĩnh vực xã hội, giáo dục... người dùng có thể truy cập các trang Web tìm kiếm khổng lồ như : <http://google.com>, <http://yahoo.com>, <http://baamboo.com> ...Từ những trang Web này, người dùng chỉ cần gõ vào từ khóa tìm kiếm, trình duyệt sẽ hiển thị tất cả những kết quả liên quan đến từ khóa chỉ trong một khoảng thời gian rất ngắn. Ngoài ra, đối với Google.com còn tích hợp những tiện ích tìm kiếm nâng cao giúp cho người sử dụng có thể tìm kiếm một cách chính xác nhất.

Hiện nay, với việc áp dụng công nghệ định vị GPRS, một số Website có thể trình diễn bản đồ thế giới một cách chính xác( điển hình như Google Earth : <http://earth.google.com> hoặc trang <http://diadiem.com> của Việt Nam hiện nay cũng được đánh giá là trang Web tìm kiếm bản đồ mạnh mẽ dành cho khu vực Thành phố Hồ Chí Minh )

Ngoài ra, các Website thông tin khác như : <http://dantri.com.vn>, <http://24h.com.vn>, <http://vietnamnet.vn> ...cùng một số các Website cá nhân, doanh nghiệp khác cũng là kho thông tin khổng lồ dành cho tất cả người sử dụng khai thác.

#### b. Nghe nhạc, chơi game, giải trí

Số lượng các trang Web nghe nhạc chiếm số lượng khá lớn trong thế giới Web. Điển hình những trang Web nghe nhạc trực tuyến và download nhạc miễn phí như : <http://nhacso.net>, <http://nhac.vui.vn>, <http://mp3.zing.vn>. Truy cập các Website này, người dùng có thể thoải mái

nghe và tải về những ca khúc mà mình yêu thích cũng như nắm bắt các thông tin về âm nhạc, nghệ thuật mà mình quan tâm.

Với việc phát triển kỹ thuật đồ họa Game 3D cùng với Internet đã bùng nổ các game trực tuyến hấp dẫn với người dùng. Điển hình những Website về Game nổi tiếng như <http://vtc.vn> , <http://trochoiviet.com> ... Những website này cũng đem lại những giây phút giải trí bổ ích cho người dùng Internet.

#### c. Học tập, nghiên cứu khoa học

Việc học tập và nghiên cứu khoa học hiện nay đã thật sự mở rộng và đến với hầu hết người dùng sử dụng Internet. Những tài liệu nghiên cứu về mọi lĩnh vực có ở khắp nơi trong các trang Web như : <http://3c.com> , <http://khoahoc.com.vn> . Về giáo dục, có thể kể đến những Website còn được gọi là Cổng thông tin điện tử (Portal) như : <http://egov.gov.vn> , <http://nghean.gov.vn> ....

Ngoài ra các diễn đàn học tập cũng là nguồn tài nguyên quý báu, ở đây người dùng có thể trao đổi , giao lưu học tập lẫn nhau trên một diễn đàn ảo và điều này thật sự là tiện lợi và bổ ích. Các sinh viên ở một số trường đại học cũng đã dựa trên nền tảng sử dụng các ngôn ngữ thiết kế Website để xây dựng những diễn đàn của trường mình nhằm mục đích quảng bá , giao lưu học tập giữa các sinh viên , điều này giúp nâng cao trình độ học vấn cũng như tình đoàn kết của tất cả sinh viên trong toàn trường. Một số diễn đàn điển hình như : <http://bka.vn> , <http://dhvinh.com>. Truy cập các diễn đàn , người sử dụng phải đăng ký 1 tài khoản với mật khẩu xác nhận và có thể tham gia gửi bài , trao đổi, bình luận những vấn đề quan tâm cùng với tất cả thành viên trên diễn đàn đó.

#### d. Kinh doanh, thương mại điện tử

Ngày nay, con người có thể không cần phải đến tận nơi để mua mặt hàng mình muốn mà có thể sử dụng dịch vụ Web để tìm kiếm mặt hàng, liên lạc với nhà cung cấp và nhận hàng ngay tại nhà. Đó là lợi ích vô cùng to lớn mà Web mang lại. Điển hình cho dịch vụ này phải kể đến các trang Web bán hàng trên mạng nổi tiếng như : <http://ebay.com>, <http://amazon.com>, ... Ở Việt Nam, mặc dù nhu cầu mua hàng trên Internet đã phát triển nhưng chỉ tập trung ở những thành phố lớn như Hà Nội , Tp.Hồ Chí Minh. Và các Website đáng tin cậy như <http://thegioididong.com> , <http://raovat.com> ... Để sử dụng dịch vụ này , người dùng có thể truy cập vào các Website trên , đăng ký tài

khoản và tiến hành trao đổi, mua bán và chuyển tiền bằng các hình thức tiền mặt, chuyển khoản....

### 1.3 Dịch vụ thư điện tử

#### 1.3.1. Khái niệm

Dịch vụ thư điện tử Email ( Electronic Mail ) là một hệ thống chuyển nhận thư từ qua các mạng máy tính.

Email là một phương tiện thông tin rất nhanh. Các thông tin ( thư từ ) có thể được gửi đi ở dạng mã hóa hay dạng thông thường và được chuyển qua các mạng máy tính đặc biệt là mạng Internet. Email có thể chuyển thông tin từ một máy nguồn đến một hay nhiều máy đích cùng một lúc.

Nội dung thông tin trong một Email rất đa dạng , có thể là hình ảnh , âm thanh hoặc là các mã HTML.

#### 1.3.2. Phần mềm thư điện tử

Là phần mềm hỗ trợ cho người dùng việc soạn thảo, đọc , in , gửi và nhận Email. Có 2 trường hợp phân biệt phần mềm thư điện tử :

Email Client : Là loại phần mềm thư điện tử được cài đặt trên các máy tính của người dùng. Ví dụ : Microsoft Outlook Express, Netscape Communicator, Eudora.

Webmail : Là phần mềm thư điện tử không cần phải cài đặt mà được cung ứng bởi các máy chủ ( Web Server ) trên Internet. Để dùng được các phần mềm loại này thì máy tính nối vào phải sử dụng một máy truy cập tương thích với sự cung ứng của Webmail. Ví dụ điển hình cho loại này là : <http://mail.yahoo.com>, <http://mail.google.com> ...

#### 1.3.3. Các giao thức sử dụng trong hệ thống Email

SMTP : Giao thức dùng cho việc gửi Email. Sử dụng port 25 của máy tính.

POP3 : Giao thức dùng cho việc nhận Email từ MailServer. Pop3 sử dụng port 110 của máy tính.

#### 1.3.4. Sử dụng Email

##### a. Sử dụng dịch vụ Gmail của Google

Google cung cấp cho người dùng dịch vụ Email miễn phí gọi là Gmail. Gmail hiện nay được đa số người sử dụng Internet tin dùng và đánh giá là

hiệu quả và chất lượng, tin cậy về tốc độ cũng như dung lượng sử dụng Email ( 7430 Mb ). Để đăng ký một tài khoản, người dùng truy cập vào địa chỉ <http://mail.google.com> để đăng ký, giao diện hoàn toàn bằng tiếng việt và dễ sử dụng. Sau khi có tài khoản, đăng nhập tài khoản và bắt đầu gửi một Email bằng cách nhấn vào **Soạn thư**. Tại đây, điền địa chỉ đầy đủ của người nhận tại mục **Tới**, sau đó ghi chủ đề cho bức thư, có thể đính kèm tệp gửi đi bằng cách nhấn vào **Đính kèm tệp**. Gmail cho phép đính kèm tệp khá nhanh và có chức năng tự động quét virus khi người nhận nhận mail. Ngoài ra Gmail cho phép tạo nhóm liên lạc để người dùng có thể cùng 1 lúc gửi 1 bức thư cho tất cả những thành viên trong nhóm một cách nhanh chóng.

Một tính năng đặc biệt của Gmail hỗ trợ cho người dùng sử dụng OutlookExpressed đó là **Chuyển tiếp POP/IMAP**. Sử dụng chức năng này, người dùng có thể cài đặt OutlookExpressed để nhận email từ Mail Server của Gmail chuyển về tại máy của mình mà không cần mở trình duyệt để truy cập.

#### b. Sử dụng dịch vụ email của Yahoo

Tiêu chí của Yahoo đó là phát triển các dịch vụ miễn phí cho người sử dụng, trong đó có dịch vụ Email. Sử dụng Yahoo mail cũng tương tự như Gmail, tuy nhiên xét về dung lượng sử dụng, tốc độ cũng như tính năng hỗ trợ thì Gmail vẫn chiếm vị trí cao hơn so với Yahoo mail.

Tuy nhiên, hiện nay Yahoo đã bắt đầu thu phí một số dịch vụ mail như : kết nối với Yahoo mail thông qua POP3 , sử dụng tính năng chuyển Yahoo mail sang tài khoản khác. Như vậy , khi người dùng sử dụng OutlookExpressed để lấy email từ Mail Server của Yahoo thì bắt buộc phải trả phí nếu không muốn bị mất luôn dịch vụ của họ.

Để sử dụng dịch vụ mail của Yahoo, truy cập website <http://mail.yahoo.com> để đăng kí 1 tài khoản. Đăng nhập và sử dụng như những phần mềm email khác.

### 1.4 Kết nối máy tính với Internet

Để kết nối máy tính với Internet phải cần có các thành phần sau : Modem của nhà cung cấp dịch vụ ( có dây hoặc không dây ). Máy tính có tích hợp NIC ( Card mạng ).

Một máy tính muốn kết nối vào mạng Internet thì cần được định danh bằng một địa chỉ IP duy nhất trên mạng.

#### 1.4.1 Kết nối máy tính với Internet bằng mạng hữu tuyến

Sử dụng Modem đã cấu hình dịch vụ Internet, dây Cable mạng, 2 đầu bấm RJ45. Bấm dây Cable mạng với chuẩn A hoặc B cho 2 đầu, 1 đầu dây Cable cắm vào khe RJ45 của Modem, đầu dây còn lại cắm vào Card mạng của máy tính. Kết nối Modem với Internet, cấu hình đúng theo nhà cung cấp dịch vụ đã cung cấp.

Nếu Modem hỗ trợ DHCP (dịch vụ cấp phát IP động) thì chỉ cần vào Run, gõ: `ncpa.cpl`, nhấp chuột phải vào **Local Area Network / Properties**, chọn **Support / Repair** để modem cấp phát IP cho máy tính truy cập vào Internet.

Hoặc vào Run gõ dòng lệnh: `cmd`. Sau đó gõ: `ipconfig/renew`.

Nếu modem không hỗ trợ DHCP thì vào Run gõ `ncpa.cpl`, nhấp chuột phải vào **Local Area Network**, chọn **Properties** / chọn **Internet Protocol (TCP/IP)**. Nhập địa chỉ IP cho máy tính với **Default Gateway** do nhà cung cấp dịch vụ cấp, ví dụ:

Ip address : 192.168.1.2

Subnet Mask : 255.255.255.0

Default Gateway : 192.168.1.1

Preferred DNS Server : 203.162.0.181

Alternate DNS Server : 203.162.4.1

#### 1.4.2. Kết nối máy tính với Internet thông qua mạng không dây (Wireless)

Khởi động chức năng wireless của máy tính

Thiết lập IP động hoặc tĩnh tương tự như mạng có dây

#### 1.4.3. Một số thủ thuật sử dụng Internet

##### 1.4.3.1 Thay đổi **Proxy, Sock**

Mục đích của việc thay đổi Proxy, Sock ngoài việc che giấu IP thật sự của máy tính thì nó còn làm cho tốc độ truy cập một số Website nhanh hơn do sử dụng cache của các ProxyServer.

Giả sử một người dùng ở Việt Nam muốn truy cập vào một số Website mà máy chủ nằm ở Trung Quốc hoặc Mỹ nhưng các máy chủ này đã thiết đặt lệnh cấm những IP từ Việt Nam truy cập vào hệ thống WebServer của họ. Việc sử dụng IP của Việt Nam để truy cập vào đã hoàn toàn bị từ chối. Trong trường hợp này, nếu sử dụng 1 Proxy của Trung Quốc hoặc Mỹ

thì IP của máy tính lúc này có định danh của Trung Quốc hoặc Mỹ và việc truy cập vào các Website đó đã hoàn toàn hợp lệ.

Các bước thay đổi Proxy :

Truy cập đường dẫn : <http://proxy4free.com> để lấy một proxy trong proxy list . Ví dụ : Proxy của China : **218.75.100.114 8080( port )**

Mở Internet Explorer, Vào **Tool/ Internet Options**. Chọn Tab **Connection / Lan settings** .Đánh dấu tích vào : **Use a proxy server for your LAN** . Nhập proxy trên vào ô text . Nhấn **OK** để đồng ý.

Kiểm tra bằng cách : mở trình duyệt, gõ vào địa chỉ : <http://google.com> . Nếu google chuyển thành <http://google.cn> là thành công.

#### 1.4.3.2 Sử dụng bảo mật trong Internet Explorer

Để cấm người dùng truy cập vào địa chỉ của một số trang web trên mạng Internet có thể sử dụng cách sau :

Mở Internet Explorer. Vào **Tool / Internet Options / Content** . Trong **Content Advisor** chọn **Enable**. Click **Setting** để cài đặt level cho các thông số cấm như : Cấm các trang web Sex, Nudity, Violence (bạo lực) lên mức 4 là mức cao nhất.

Chọn Tab **Approved Sites** , gõ địa chỉ website cần cấm vào , nhấn **Nerver** để cấm truy cập. ( ví dụ : <http://dantri.com.vn> )

**Câu 2: Các phương pháp tìm kiếm trong ( tìm kiếm tuần tự, tìm kiếm nhị phân, cây nhị phân tìm kiếm )**

### **A. GIỚI THIỆU CHUNG**

Tìm kiếm là một thao tác rất quan trọng đối với nhiều ứng dụng tin học. Tìm kiếm có thể định nghĩa là việc thu thập một số thông tin nào



đó từ một khối thông tin lớn đã được lưu trữ trước đó. Thông tin khi lưu trữ thường được chia thành các bản ghi, mỗi bản ghi có một giá trị khoá để phục vụ cho mục đích tìm kiếm. Mục tiêu của việc tìm kiếm là tìm tất cả các bản ghi có giá trị khoá trùng với một giá trị cho trước. Khi tìm được bản ghi này, các thông tin đi kèm trong bản ghi sẽ được thu thập và xử lý.

Một số ví dụ về ứng dụng thực tiến của tìm kiếm là từ điển máy tính. Trong từ điển có rất nhiều mục từ, khoá của mỗi mục từ chính là cách viết của từ. Thông tin đi kèm là định nghĩa của từ, cách phát âm, các thông tin khác như loại từ, từ đồng nghĩa, khác nghĩa v.v. Ngoài ra còn rất nhiều ví dụ khác về ứng dụng của tìm kiếm, chẳng hạn một ngân hàng lưu trữ các bản ghi thông tin về khách hàng và muốn tìm trong danh sách này một bản ghi của một khách hàng nào đó để kiểm tra số dư và thực hiện các giao dịch, hoặc một chương trình tìm kiếm duyệt qua các tệp văn bản trên máy tính để tìm các văn bản có chứa các từ khoá nào đó.

Bài toán tìm kiếm bao gồm hai nhóm : tìm kiếm nội và tìm kiếm ngoại. Nếu lượng dữ liệu lớn phải lựa trên thiết bị lưu trữ ngoài như đĩa hay băng từ thì bài toán được gọi là tìm kiếm ngoại. Ngược lại nếu toàn bộ dữ liệu được lựa trên bộ nhớ chính thì được gọi là tìm kiếm nội. Ở đây chúng em chỉ quan tâm đến tìm kiếm nội.

## **B. TÌM KIẾM TUẦN TỰ**

Tìm kiếm tuần tự là một phương pháp tìm kiếm đơn giản, lần lượt duyệt qua toàn bộ các bản ghi một cách tuần tự. Tại mỗi bước, khoá của bản ghi sẽ được so sánh với giá trị cần tìm. Quá trình tìm kiếm kết thúc khi đã tìm thấy bản ghi có khoá thoả mãn hoặc đã duyệt hết danh sách.

### **a. Mô tả bài toán tìm kiếm**

“Cho 1 bảng chính gồm  $n$  bản ghi  $R_1, R_2, \dots, R_n$ . Mỗi bản ghi  $R_i$  tương ứng với một khoá  $K_i$  ( $1 \leq i \leq n$ ). Hãy tìm bản ghi có giá trị khoá tương ứng bằng  $X$  cho trước”.

- $X$  được gọi là khoá tìm kiếm hay đối trị tìm kiếm.
- Công việc tìm kiếm sẽ hoàn thành khi xảy ra một trong hai tình

huống sau :

- + Tìm được bản ghi có giá trị khoá =  $X$  (thành công)
- + Không tìm được bản ghi có giá trị khoá =  $X$  (không thành công)

Sau một phép tìm kiếm không thành công, có thể xuất hiện yêu cầu bổ sung thêm bản ghi mới, giải thuật như vậy được coi là “giải thuật tìm kiếm có bổ sung”.

Giá trị của khoá có thể là số, ký tự, xâu ký tự....( đối với bài toán trên ta coi khoá là các số nguyên).

### b. Nguyên tắc tìm kiếm

- Lần lượt so sánh  $X$  (khoá tìm kiếm) với các khoá  $K_1, K_2, \dots, K_n$  trong bảng cho tới khi tìm thấy  $X$  ( $X = K_m$ ) hoặc hết bảng mà chưa tìm thấy  $X$ .
- Kết quả :
  - + Tìm được vị trí  $m$  của khoá (Đầu tiên) có giá trị bằng  $X$ .
  - + Không tìm được khoá có giá trị bằng  $X$ .

### c. Giải thuật

Ở bài toán này mỗi phần tử  $R_1, R_2, \dots, R_n$  chúng ta sẽ dùng kiểu dữ liệu cấu trúc (struct) để biểu diễn và dữ liệu trong mỗi bản ghi thì gồm có hai phần :

- Khoá (với kiểu dữ liệu là (int) )

- nội dung (nội dung của bản ghi với kiểu dữ liệu là (char) )

Cho bảng gồm  $n$  phần tử ( $n \geq 0$ ), các khoá và  $X$  là các số nguyên (int) được nhập từ bàn phím, nội dung của mỗi phần tử là một xâu kí tự (char). Giải thuật tìm kiếm tuần tự sẽ thực hiện tìm kiếm trong bảng xem có khoá nào bằng  $x$  không.

+ nếu có sẽ đưa ra chỉ số của khoá ấy

+ Nếu không nó sẽ đưa ra giá trị là 0

+ Trường hợp không tìm thấy thì sẽ xuất hiện yêu cầu bổ sung phần tử mới có khoá là giá trị vừa đưa ra để tìm kiếm và nội dung thì chúng ta sẽ nhập vào (từ bàn phím ).

+ Trong giải thuật này có sử dụng một khoá phụ  $K_{n+1}$  mà giá trị của nó chính là  $X$ .

#### **Mô tả chi tiết về các bước của giải thuật:**

Bước 1: Đầu tiên nhập và lưu  $N$  phần tử và khoá  $X$  cần tìm (Giữ liệu được lưu vào mảng).

Bước 2: Gán  $i=1$  có nghĩa xét số hạng đầu tiên trong dãy.

Bước 3: Kiểm tra xem  $K_i$  có bằng  $X$  không. Lúc này ta đang kiểm tra

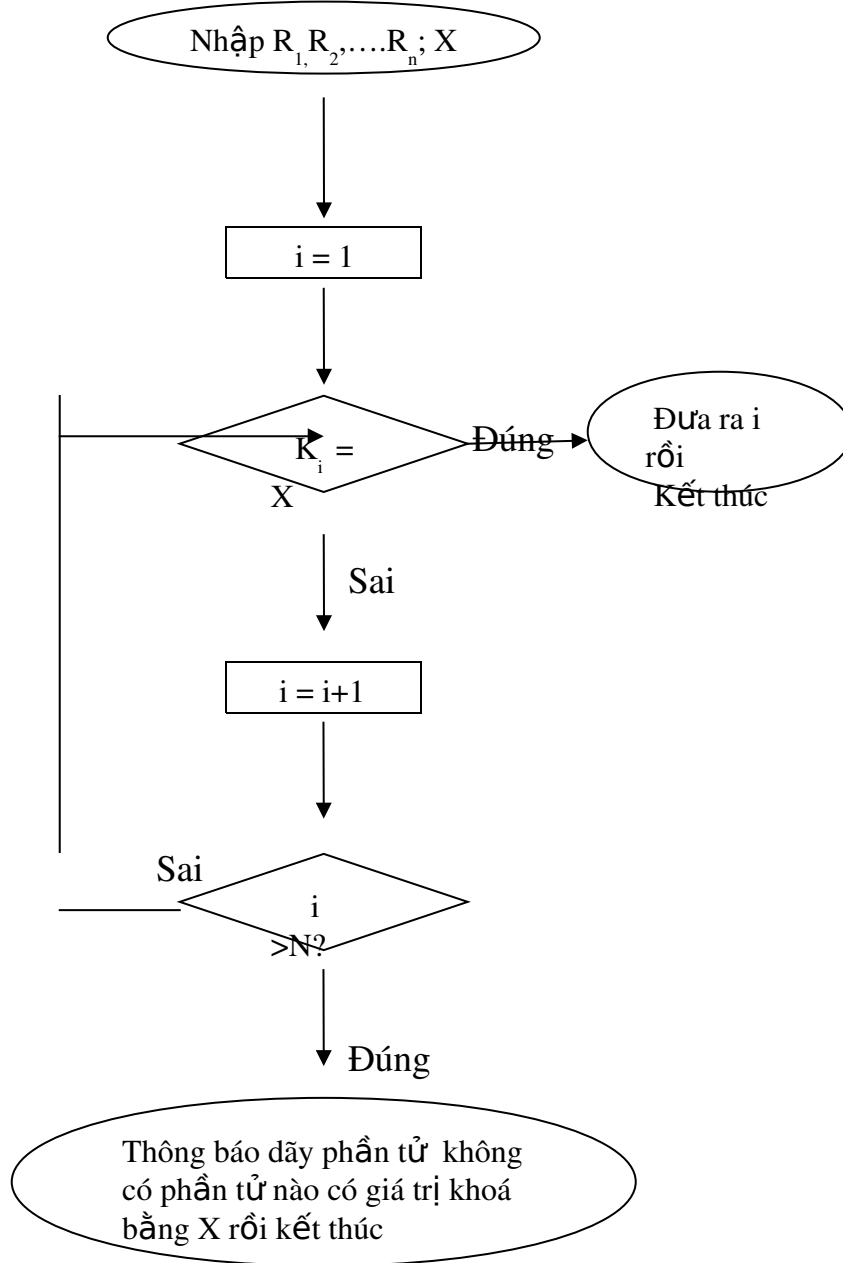
$K_i=X$ ? nếu bằng thì đã tìm thấy thông báo ra chỉ số  $i$  và dừng thuật toán ngược lại ta sang bước 4.

Bước 4: Ta kiểm tra số hạng tiếp theo bằng cách tăng  $i = i+1$ ;

Nếu số hạng tiếp theo mà có chỉ số lớn hơn  $N(i < n)$  tức là dãy đã được kiểm tra hết mà không có phần tử nào bằng khoá. Lúc này ta dừng thuật toán và thông báo không tìm thấy  $i$  để  $K_i=X$ ;

Ngược lại  $i < N$  thì ta tiếp tục quay lại bước 3 để tiến hành tìm kiếm.

**Sơ đồ khối:**



Sau đây là **giải thuật**(viết theo ngôn ngữ C) :

```

Int timkiem (k,n,x )
{
    int i=1 ;
    // Có thể đặt khoá phụ để bổ sung thêm phần tử này khi
    không
        tìm thấy nó trong dãy khoá
    K[n+1] = X;
    // Tìm kiếm tuần tự
    While ((i <= n +1)&&(K[i] != X))
        i = i + 1;
    // không tìm thấy
    if (i == n+1)
        Return 0;
    else
        // tìm thấy
        Return i;
}
    
```

**Ví dụ minh họa:**

Cho dãy các phần tử với các giá trị khoá được cho tương ứng như sau:

K	3	5	6	7	8
i	1	2	3	4	5

--	--	--	--	--	--

Trong đó với mỗi giá trị K có các chỉ số i tương ứng ghi phía dưới.  
 Hãy tìm xem trong dãy trên có phần tử nào có khoá bằng 7 không.  
 Nếu có thì đưa ra chỉ số của phần tử đó.

Bước 1: Nhập các phần tử và khoá X cần tìm( $X = 7$ )

Bước 2: Gán  $i = 1$

Bước 3:  $K[1] = 3 \neq X$  chuyển sang bước 4

Bước 4:  $i = i + 1 = 2 < 5$  quay lại bước 3

-  $K[2] = 5 \neq X$  chuyển sang bước 4

-  $i = i + 1 = 3 < 5$  quay lại bước 3

-  $K[3] = 6 \neq X$  chuyển sang bước 4

-  $i = i + 1 = 4 < 5$  quay lại bước 3

-  $K[4] = X$ . Vậy là đã tìm thấy trả về chỉ số i và kết thúc, thuật toán đã thành công.

Code:

```
#include<conio.h>
#include<stdio.h>
#include<string.h>
typedef struct phantu
{
    int khoa;
    char noidung[40];
}phantu;
phantu p[50];
int n;
void bosung()
{
    n++;
    printf("\n\nBan nhap noi dung can bo sung:");
    scanf("%s",&p[n-1].noidung);
}
void inds()
{
    int i;
    printf("\n\n danh sach sau khi da nhap :");
    for(i=0;i<n;i++)
```

```

    {
        printf("\n\n khoa %d",p[i].khoa);
        printf("\n noi dung : %s",p[i].noidung);
    }

}

int timkiem(int kh)
{
    int i;
    p[n].khoa=kh;
    for(i=0;i<n;i++)
    {
        if(p[i].khoa==kh)
            return i+1;
    }
    return -1;
}

int tontai(int so)
{
    for(int i=0;i<n;i++)
        if(p[i].khoa==so)return 1;
    return 0;
}

void tao()
{
    int so;
    n=0;
    do
    { nhaplai:
        printf("\n Nhap khoa : ");
        scanf("%d",&so);
        if(so>0)
        { if(!tontai(so))
            {
                p[n].khoa=so;
                printf("\n nhap noi dung :");
                scanf("%s",&p[n].noidung);
            }
        }
    } while(so>0);
}

```

```

        n++;
    }
    else
    {
        printf("\n\nBan nhap trung khoa. ");
        goto nhaplai;
    }
}
}while(so>0);

}
void main()
{
    clrscr();
    int k;
    char ch;
    tao();
    inds();
    printf("\n\n\n nhap khoa can tim: ");
    scanf("%d",&k);
    int l=timkiem(k);
    if(l!=-1)
        printf("\n\n tim thay o vi tri thu %d",l);

    else
    {
        printf("\n\n khong tim thay!");
        printf("\n\nBan co muon bo sung them phan tu co gia tri
khoa = %d khong(C/K)?",k);
        ch=getch();
    }
    if(ch=='c')
    {        bosung();
        inds();
    }

    getch();
}

```



}  
**C. TÌM KIẾM NHỊ PHÂN**

**a. Mô tả bài toán tìm kiếm**

“ Cho 1 bảng chính gồm n bản ghi  $R_1, R_2, R_3, \dots, R_n$ . Mỗi bản ghi  $R_i$  tương ứng với một khoá  $K_i$  ( $1 \leq i \leq n$ ). Hãy tìm bản ghi có giá trị khoá tương ứng bằng X cho trước”.

- X được gọi là khoá tìm kiếm hay đối trị tìm kiếm.
- Công việc tìm kiếm sẽ hoàn thành khi xảy ra một trong hai tình huống

sau:

- + Tìm được bản ghi có giá trị khoá = X (thành công)
- + Không tìm được bản ghi có giá trị khoá = X (không thành công)

Sau một phép tìm kiếm không thành công, có thể xuất hiện yêu cầu bổ sung thêm bản ghi mới, giải thuật như vậy được coi là “giải thuật tìm kiếm có bổ sung”.

Giá trị của khoá có thể là số, ký tự, xâu ký tự.....( đối với bài toán trên ta coi khoá là các số nguyên).

**b. Nguyên tắc tìm kiếm**

Phép tìm kiếm nhị phân luôn chọn khoá “ở giữa” bảng khoá đang xét để thực hiện so sánh với khoá tìm kiếm x.

Nguyên tắc tìm kiếm :

- Giả sử bảng khoá đang xét là  $K_1, \dots, K_R$  thì khoá ở giữa bảng sẽ là  $K_i$

với :  $i = [(L + R)/2]$ . (lấy phần nguyên)

- Tìm kiếm sẽ kết thúc nếu  $X = K_i$ .
- Nếu  $X < K_i$  tìm kiếm được thực hiện tiếp với  $K_L, \dots, K_{i-1}$ .
- Nếu  $X > K_i$  tìm kiếm được thực hiện tiếp với  $K_{i+1}, \dots, K_R$ .
- Quá trình tìm kiếm bắt đầu với  $L = 1$  và  $R = n$  đến khi tìm thấy X hoặc bảng khoá đang xét rỗng (không tìm thấy).

**c. Giải thuật**

Cho bảng gồm  $n$  phần tử ( $n$  là số nguyên nhập từ bàn phím) đã được sắp xếp theo thứ tự tăng dần (Sắp xếp căn cứ vào việc so sánh giá trị các khoá giữa các phần tử với nhau).

Giải thuật tìm kiếm nhị phân sẽ được thực hiện tìm kiếm trong bảng xem có khoá nào bằng  $x$  không. Nếu có sẽ đưa ra chỉ số của khoá ấy, còn nếu không nó sẽ đưa ra giá trị 0.

Trường hợp không có sẽ xuất hiện yêu cầu bổ sung thêm phần tử mới có khoá là giá trị vừa đưa ra để tìm kiếm và nội dung thì chúng ta sẽ nhập vào (từ bàn phím).

**Mô tả chi tiết về các bước của giải thuật:**

Bước 1: Nhập và lưu các phần tử và khoá  $X$  cần tìm.

Bước 2:  $Dau = 1, Cuoi = N$

Bước 3:  $Giua = [(Dau+Cuoi)/2]$

Bước 4: Nếu  $K_{Giua} = X$  thì thông báo chỉ số  $Giua$ , rồi kết thúc.

Bước 5: Nếu  $K_{Giua} > X$  thì  $Cuoi = Giua - 1$ , rồi chuyển đến bước

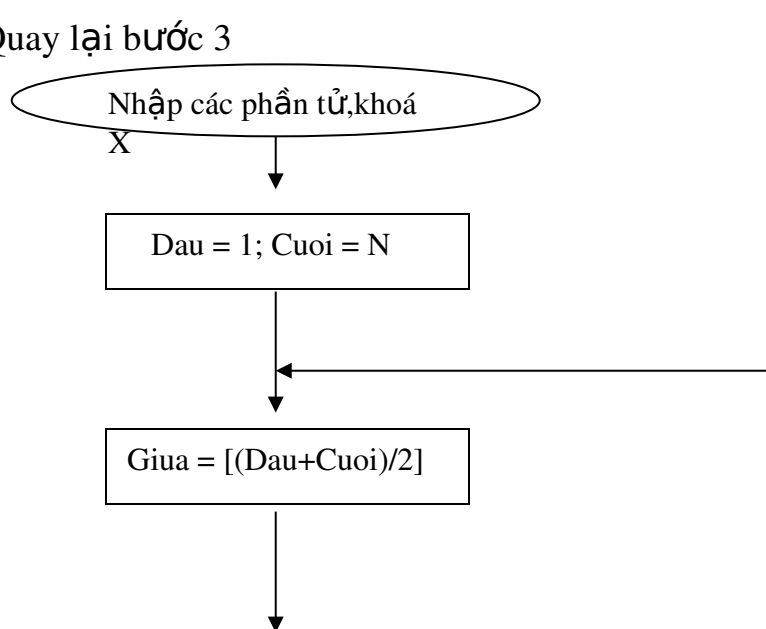
7

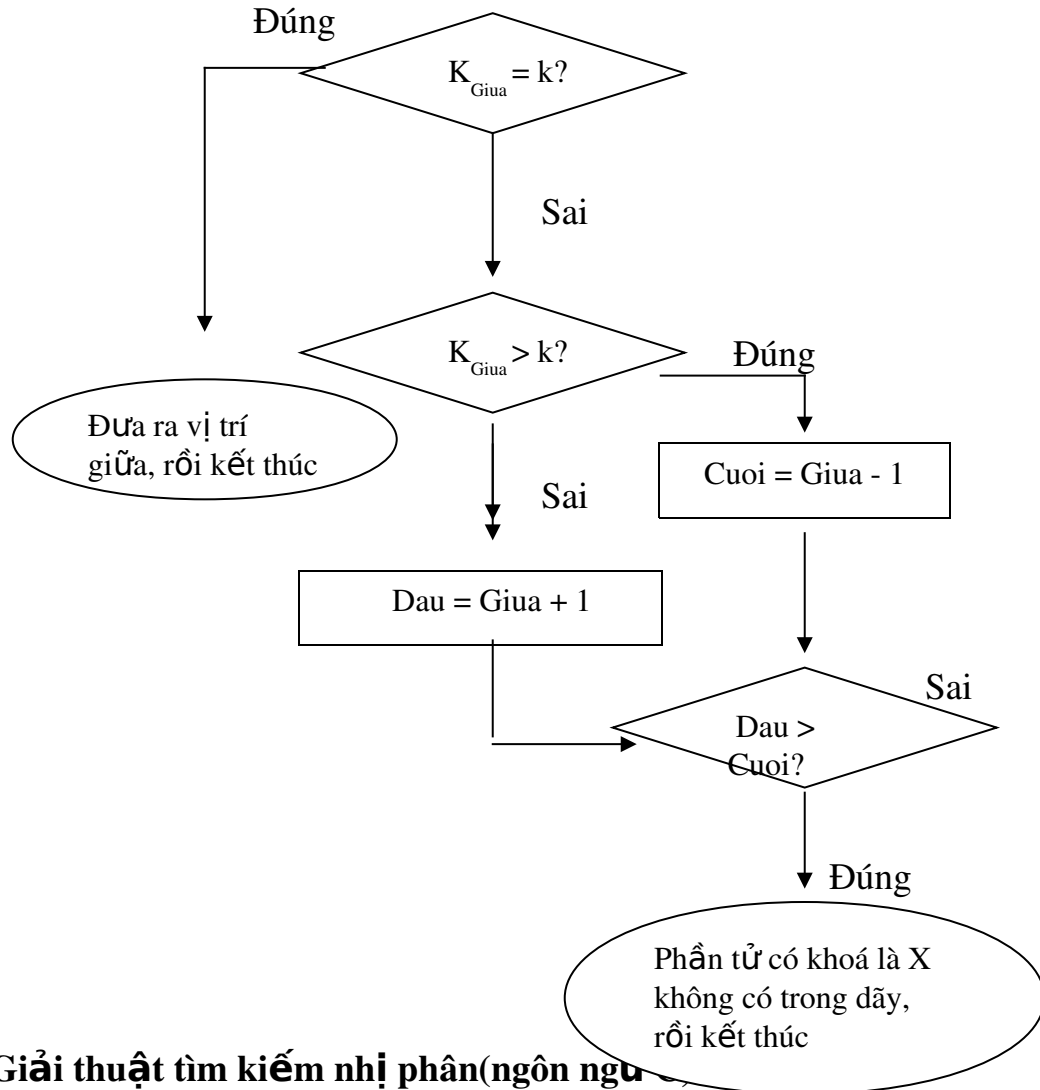
Bước 6:  $Dau = Giua + 1$

Bước 7: Nếu  $Dau > Cuoi$  thì thông báo dãy không có phần tử nào có giá trị khoá bằng  $X$ , rồi kết thúc.

Bước 8: Quay lại bước 3

**Sơ đồ khối:**





**Giải thuật tìm kiếm nhị phân (ngôn ngữ C)**

```

Int timkiem( k, kh, n)
{
// L là chỉ số phần tử đầu tiên, còn r chỉ số phần tử cuối cùng
    Int l=1,r=n,m;
    While(l<=r)
    {
        m=(l+r)/2;

        // Tìm ở nửa đầu của dãy khoá
        if(k[m]>kh)
            r=m-1;
    }
}
    
```

```

else

// Tìm Ở nửa sau của dãy khoá
if(k[m]<kh)
    l=m+l;
else

// Đã tìm thấy (Phần tử Ở giữa trùng với khoá cần tìm)
return m;
}

// Không tìm thấy
Return 0;
}

```

**Ví dụ minh hoạ:**

Cho dãy các phần tử với các giá trị khoa được cho tương ứng như sau:

K	1	3	4	6	7	10	15
i	1	2	3	4	5	6	7

Trong đó với mỗi giá trị K có các chỉ số i tương ứng ghi phía dưới.

Hãy tìm xem trong dãy trên có phần tử nào có khoá bằng 15 không.  
Nếu có thì đưa ra chỉ số của phần tử đó.

Các bước giải được liệt kê theo bảng dưới đây:

○	K	1	3	4	6	7	10	15
	Đầu	1	5	7				
	Cuối	7	7	7				
○	Giữa	4	6	7				
	$K_{Giua}$	6	10	15				

**Kết quả:** Tìm thấy khoá X = 15 tại vị trí Giữa = 7 trong dãy

Code:

```
#include<conio.h>
#include<stdio.h>
#include<string.h>
typedef struct phantu
{
    int khoa;
    char noidung[30];
}phantu;
phantu p[50];
int n;
void bosung(int k)
{
    n++;
    p[n-1].khoa=k;
```

```

printf("\n\nBan nhap noi dung can bo sung:");
scanf("%s",&p[n-1].noidung);
}
void inds()
{   int i;
    for(i=0;i<n;i++)
    {
        printf("\n\n khoa %d",p[i].khoa);
        printf("\n\t noi dung : %s",p[i].noidung);
    }
}
void sapxep()
{
    phantu tg;
    for(int i=0;i<n-1;i++)
        for(int j=i+1;j<n;j++)
            if(p[i].khoa>p[j].khoa)
            {
                tg=p[i];
                p[i]=p[j];
                p[j]=tg;
            }
}
int timkiem(int kh)
{
    int l=0,r=n,m;
    sapxep();
    while(l<=r)
    {
        m=(l+r)/2 ;
        if(p[m].khoa>kh)
            r=m-1;
        else
            if(p[m].khoa<kh)
                l=m+1;
        else
            return m;
    }
}

```

```

    }
    return 0;
}

int tontai(int so)
{
    for(int i=0;i<n;i++)
        if(p[i].khoa==so)return 1;
    return 0;
}

void tao()
{
    int so;
    n=0;
    do
    { nhaplai:
        printf("\n Nhap khoa : ");
        scanf("%d",&so);
        if(so>0)
        {   if(!tontai(so))
            {
                p[n].khoa=so;
                printf("\n nhap noi dung :");
                scanf("%s",&p[n].noidung);
                n++;
            }
            else
            {
                printf("\n\nBan nhap trung khoa. ");
                goto nhaplai;
            }
        }
    }while(so>0);
}

void main()

```

```

{
    clrscr();
    int k,l;
    char ch;
    tao();
    printf("\n\n\n danh sach sau khi da nhap: ");
    inds();
    sapxep();
    printf("\n\n\n danh sach sau khi sap xep: ");
    inds();
    printf("\n\n\n nhap khoa can tim: ");
    scanf("%d",&k);
    l=timkiem(k);
    if(l!=0)
        printf("\n\n tim thay o vi tri thu %d",l+1);

    else
    {
        printf("\n\n khong tim thay!");
        printf("\n\nBan co muon bo sung them phan tu co gia tri
khoa = %d khong(C/K)?",k);
        ch=getch();
    }
    if(ch=='c')
    {
        bosung(k);
        inds();
    }

    getch();
}

```

#### D. Cây nhị phân tìm kiếm



**a. Các khái niệm cơ bản**

Để đáp ứng yêu cầu tìm kiếm đối với dãy khoá luôn luôn biến động, một phương pháp khác được hình thành dựa trên cơ sở dãy khoá được tổ chức dưới dạng cây nhị phân gọi là cây nhị phân tìm kiếm.

Vậy thế nào là cây nhị phân tìm kiếm?

Cây nhị phân tìm kiếm ứng với  $n$  khoá  $K_1, K_2, \dots, K_n$  là một cây nhị phân mà mỗi nút được gán một giá trị khoá sao cho mọi nút trên cây đều thoả mãn các tính chất sau :

- Mọi khoá thuộc cây con trái của một nút đều có giá trị nhỏ hơn khoá ở nút đó.
- khoá thuộc cây con phải của một nút đều có giá trị lớn hơn khoá ở nút đó.

**Nút gốc :** là nút trên cùng trong một cây, trên nút gốc không có nút nào nữa.

**Nút con :** nút kế tiếp (phía dưới) của một nút trong cây, một nút có thể có nhiều nhất là hai nút con. Các nút con này được đánh dấu theo thứ tự từ trái qua phải, nút con tận cùng bên trái là nút đầu tiên và nút con tận cùng bên phải là nút tận cùng.

**Nút cha :** nút liền kề phía trên của một nút trong cây, mỗi một nút chỉ duy nhất một nút cha.

**Các nút anh em :** các nút con của cùng một nút.

**Các cạnh, nhánh :** đường nối từ nút cha đến các nút con của nó.

**Nút tổ tiên :** là các nút phía trên của các nút con trong cây, nút gốc không có nút tổ tiên.

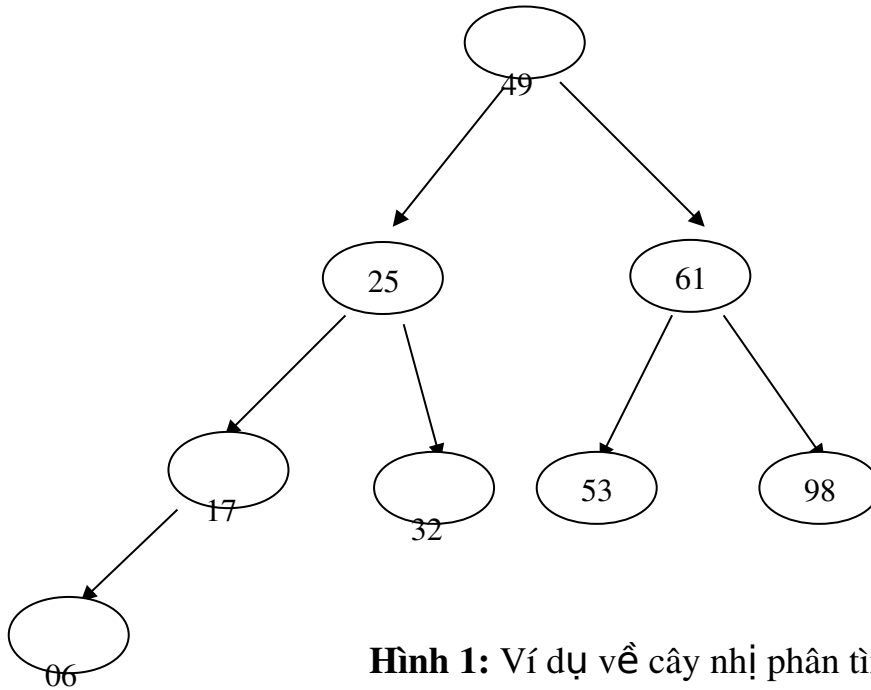
**Đường đi :** là chuỗi các cạnh nối từ một nút tới một trong các nút hậu duệ nào trong cây.

**Chiều dài đường đi :** số cạnh trong đường đi.

**Nút lá :** nút không có nút con.

**Chiều cao của cây :** chiều dài đường đi dài nhất trong cây.

VD : Biểu diễn dãy khoá sau thành cây nhị phân tìm kiếm.  
49 25 61 17 53 32 98 06



**Hình 1:** Ví dụ về cây nhị phân tìm kiếm

### b. Giải thuật tìm kiếm

- So sánh X với khoá ở gốc khi đó một trong 4 tình huống sau đây sẽ xuất hiện.
- Không có gốc ( Cây rỗng ): X không có trên cây, phép tìm kiếm không được thoả mãn.
- X trùng với khoá ở gốc phép tìm kiếm được thoả mãn.
- X nhỏ hơn khoá ở gốc: phép tìm kiếm được thực hiện tương tự trên cây con trái.
- X lớn hơn khoá ở gốc: phép tìm kiếm được thực hiện tương tự trên cây con phải.

Giả sử mỗi nút của cây nhị phân tìm kiếm có dạng

<b>LPR</b>	<b>DATA</b>	<b>RPTR</b>
------------	-------------	-------------

Trong đó:

- Trường DATA thì bao gồm :
  - + Trường khoá (KEY).
  - + Trường chứa các thông tin khác( nội dung INFO).
- Trường LPTR và RPTR là con trỏ trỏ đến cây con trái và cây con phải.

Cho cây nhị phân tìm kiếm có gốc trỏ bởi T. Giải thuật tìm kiếm sau sẽ trả về địa chỉ nút có giá trị khoá = X, nếu không tìm thấy thì thực hiện bổ sung vào cây.

### - Giải thuật tìm kiếm

Node \*timkiem ( T,X )

```
{
    {Khởi tạo}
    P = T;
    // P trỏ vào cây, dùng P để dịch chuyển và kiểm tra
    {Tìm kiếm}
    If(p!=NULL)
    {
        If (X < P->KEY)
            // Tìm kiếm bên trái, dùng đệ quy
            Return timkiem(p->left,X);
        Else
            If(X>p->KEY)
                //Tìm kiếm bên phải, dùng đệ quy
                Return timkiem(p->right,X);
    }
}
```

```

Else
    Return p;
//Trả kết quả tìm kiếm được về cho hàm
Return p;
}

```

- **Trường hợp tìm không thấy sẽ bổ sung(gọi hàm bổ sung)**

```

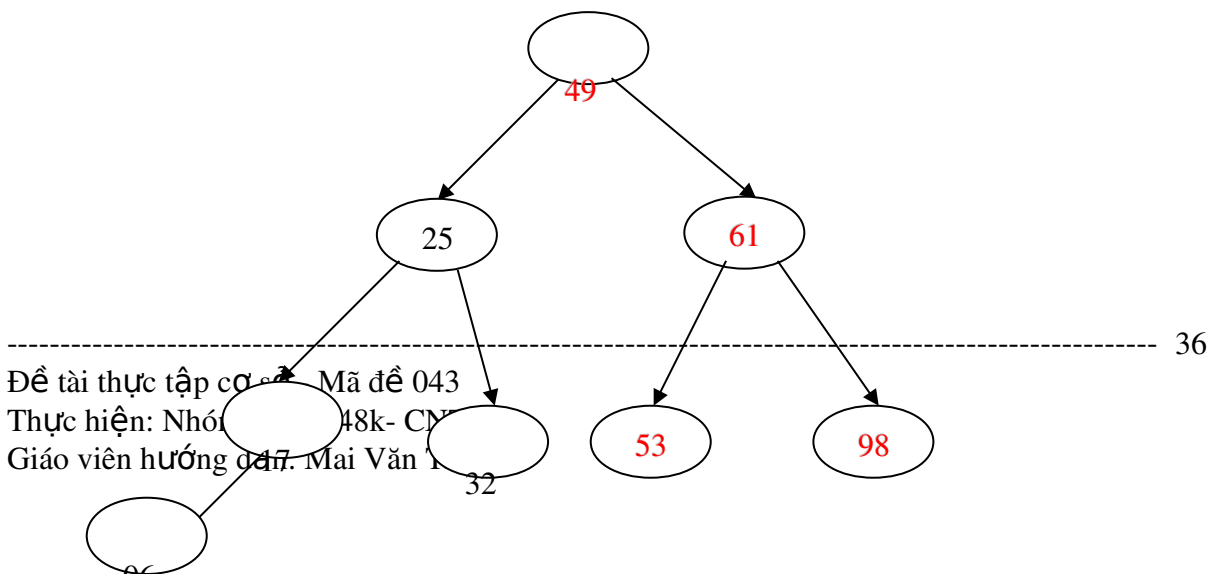
Void insert(node *&T,int X )
{
    If(T == NULL)
    {
        T = new node;
        T->KEY = X;
        T->right = T->left = NULL;
        Return;
    }
//sử dụng đệ quy để bổ sung
    If( X < T -> KEY)
        Insert(T ->left, X);
    Else
        Insert(T ->right, X);
}

```

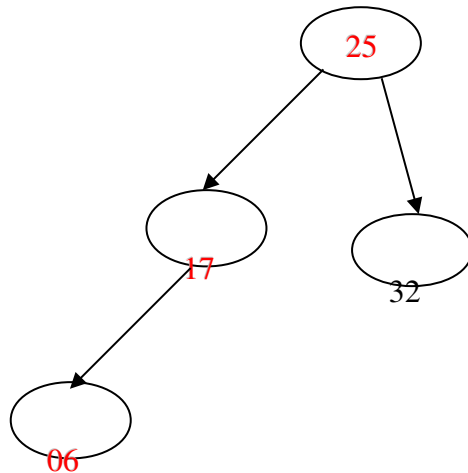
**Ví dụ minh họa:**

Xét cây nhị phân tìm kiếm như ở hình 1. Giả sử ta cần tìm nút 32 trên cây này. Quá trình tìm kiếm như sau:

Bước 1: So sánh với nút gốc là 49. Do 32 nhỏ hơn 49 nên tiến hành tìm kiếm ở cây con bên trái.



Bước 2: So sánh 32 với nút gốc của cây tìm kiếm hiện tại là 25. Do 32 lớn hơn 25 nên tiến hành tìm kiếm ở cây con bên phải.



Bước 3: So sánh 32 với nút gốc của cây tìm kiếm hiện tại cũng là 32. Do 2 giá trị bằng nên quá trình tìm kiếm kết thúc thành công.

Như vậy chỉ sau 3 phép so sánh, thao tác tìm kiếm trong 1 danh sách gồm 7 phần tử đã kết thúc và cho kết quả.

Cài đặt thuật toán :

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

```

#include <math.h>
struct tree
{
    int key;
    tree *left;// cau truc tu tro tu dong chen
    tree *right;
};
typedef tree node;
node *root;
void khoitao(node *&root)
{
    root=NULL;
}

void insert_node(node *&root,int x)
{
    if (root==NULL)
    {
        root=new node;
        root->key=x;
        root->left=NULL;
        root->right=NULL;
        return ;
    }
    if(x<root->key)
        insert_node(root->left,x);
    else
        insert_node(root->right,x);
}

void input_tree(node *&root)
{
    khoitao(root);
    int n,x;
    printf("\nNhap so nut cua cay : ");
    scanf("%d",&n);
    for(int i=0;i<n;i++)
    {

```

```

        printf("Node[%d]= ",i);
        scanf("%d", &x);
        insert_node(root,x);
    }
}
node *search_node(node *root,int x)
{
    node *p;
    p=root;
    if(p)
        if(x<p->key)
            return search_node(p->left,x);
        else
            if(x>p->key)
                return search_node(p->right,x);
            else
                return p;
    return p;
}
void output_tree(node *&root)
{
    if (root!=NULL)
    {
        printf("\t%d",root->key);
        output_tree(root->left);
        output_tree(root->right);
    }
}
int menu()
{
    int cv;
    clrscr();
    do
    {
        printf("\n BAN CHON CONG VIEC CAN LAM");
        printf("\n So 1 la tao cay nhi phan.");
    }
}

```

```

    printf("\n So 2 la duyet cay trai hay duyet truoc.");
    printf("\n So 3 la hay nhap vao khoa can tim kiem.");
    printf("\n So 0 la thoat khoi chuong trinh.\n");
    printf("\n Chon cac thao tac tren:");
    scanf("%d",&cv);
}while(cv<0||cv>4);
return cv;
}
void main()
{
    int x,cv ;
    char ch,ch1;
    do{
        cv=menu();
        switch(cv)
        {
            case 0:
            {
                clrscr();
                printf("\n chuong trinh se ket thuc!");
                getch();
                return;
            }
            case 1:
            {
                clrscr();
                input_tree(root) ;
                break;
            }
            case 2:
            {
                clrscr();
                printf("\nCay kieu tien to :\n\n");
                output_tree(root);
                getch();
                break;
            }
        }
    }
}

```



```

case 3:
{
    clrscr();
    printf("\n\n nhap khoa can tim : ");
    scanf("%d",&x);
    if (search_node(root,x) != 0)
        printf("\nPhan tu %d co trong cay .\n",x );
    else
    {
        printf("\nPhan tu %d khong co trong cay .\n",x);
        //printf("\nChen phan tu %d vua nhap vao cay .\n" ,x);
        printf("\nBan co muon bo sung them phan tu %d vau nhap vao cay
khong(C/K)?",x);
        ch1=getch();
    }
    if(ch1=='c')
    {
        insert_node(root,x);
        printf("\n\nCay nhi phan vua chen vua chen vao la .\n");
        printf("\nIn theo kieu tien to :\n\n");
        output_tree(root) ;
    }
}
printf("\nnhan ESC de ket thuc va phim bat ky de tiep tuc!!");
ch=getch();
}
}while(ch!=27);
getch();
}

```

## **Câu 3: Tìm hiểu và sử dụng Outlook Express**

### **3.1. Giới thiệu về Outlook Express**

Email (Thư điện tử) sử dụng mạng máy tính để phát và thu các thông báo. Được liên kết bởi các ghép nối dữ liệu tốc độ cao vượt qua các biên giới quốc gia, thư điện tử cho phép bạn thảo các thông báo và gửi chúng, chỉ trong vài giây, đến nhiều người nhận trong cơ quan của bạn, đến cơ quan ở bang khác, hoặc đến một địa điểm nào đó trên thế giới. Người nhận không bắt buộc phải có mặt lúc nhận thông báo, sẽ có một thông báo hiển thị trên màn hình "Bạn có một thư đang đợi" khi người nhận ghép nối với hệ thống vào một lúc nào đó.

Outlook Express(OE) là chương trình tích hợp sẵn trong window dùng để gửi và nhận Email, với OE chỉ cần mở chương trình và nhấn một nút là có thể gửi và nhận tất cả các Email một cách nhanh chóng, các Email nhận được sẽ lưu trong máy vi tính và có thể xem bất cứ lúc nào.

Điều kiện sử dụng OE để Email là phải đăng ký dịch vụ Email hỗ trợ POP3, mà điều này không khó lắm vì hiện nay khi đăng ký sử dụng ADSL thường sẽ được tặng kèm theo một hoặc nhiều địa chỉ Email và các địa chỉ này đều có thể sử dụng được bằng OE.

### 3.2. Giới thiệu Các thuật ngữ

SMTP (Simple Mail Transfer Protocol: Giao thức chuyển thư điện tử đơn giản): là một cơ chế trao đổi thư trên Internet. Nó có trách nhiệm chuyển thông điệp từ một Mail server (máy chủ chuyên lo về dịch vụ thư tín điện tử) này đến mail server khác.

Mail server chạy một giao thức kiểm soát thông điệp gọi là POP (Post Office Protocol: giao thức bưu điện) hay IMAP (Internet Mail Access Protocol: giao thức truy cập thư Internet).

POP lưu trữ các thư trong hộp thư của người sử dụng. Khi người sử dụng kết nối vào mail server thì tên thư điện tử của họ được dùng để xác minh họ là ai và cho phép họ truy cập vào hộp thư của họ. Những thông điệp sẽ được tải xuống máy của họ.

IMAP4 cải tiến mô hình này bằng cách cho phép người sử dụng giữ thư điện tử trong những hộp thư riêng trên mail server thay vì các thư này tự động được tải xuống máy của họ. Điều này có ích cho người sử dụng ở những nơi xa xôi.

SMTP giống như người mang thư có trách nhiệm chuyển thư trong khi POP và IMAP4 giống như các bưu điện có trách nhiệm nhận, lưu trữ và chuyển tiếp thư.

### 3.3. Một số thông tin thiết đặt khác đối với Mail Server của Gmail

a. Google gmail :

Incoming Mail Server (POP3) - pop.gmail.com (bật SSL, port 995).

Outgoing Mail Server - smtp.gmail.com (bật SSL, port 465).

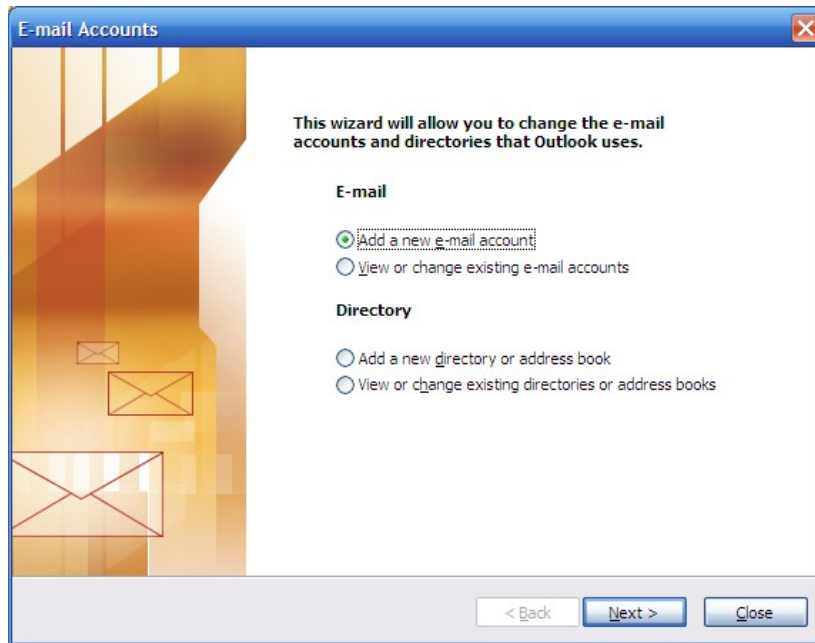
b. Mail.yahoo.com.vn:

Incoming Mail Server (POP3) - pop.mail.yahoo.com.vn (bật SSL, port 587)

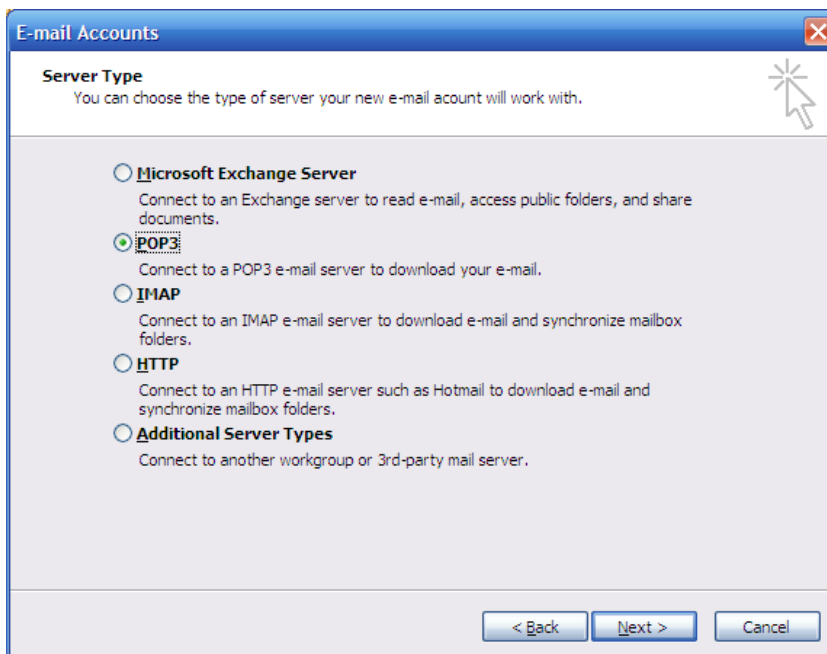
Outgoing Mail Server- smtp. mail.yahoo..com.vn (bật SSL, port 25)

### 3.4. Cài đặt hộp thư cố định trong Outlook Express

Chạy Outlook Express : Start/ Programs/ Microsoft Office/ Microsoft outlook 2003



Chọn Next



## chọn POP3

E-mail Accounts

**Internet E-mail Settings (POP3)**  
Each of these settings are required to get your e-mail account working.

**User Information**

Your Name: Hoang Thuy  
E-mail Address: tot.a.one2004@gmail.com

**Server Information**

Incoming mail server (POP3): pop.gmail.com  
Outgoing mail server (SMTP): smtp.gmail.com

**Logon Information**

User Name: tot.a.one2004  
Password: \*\*\*\*\*  
 Remember password  
 Log on using Secure Password Authentication (SPA)

**Test Settings**

After filling out the information on this screen, we recommend you test your account by clicking the button below. (Requires network connection)

Test Account Settings ...

More Settings ...

< Back   Next >   Cancel

Điền tên của bạn vào ô '**Your name**'

Điền địa chỉ đầy đủ của tài khoản Gmail của bạn (username@gmail.com) vào ô '**Email address:**'

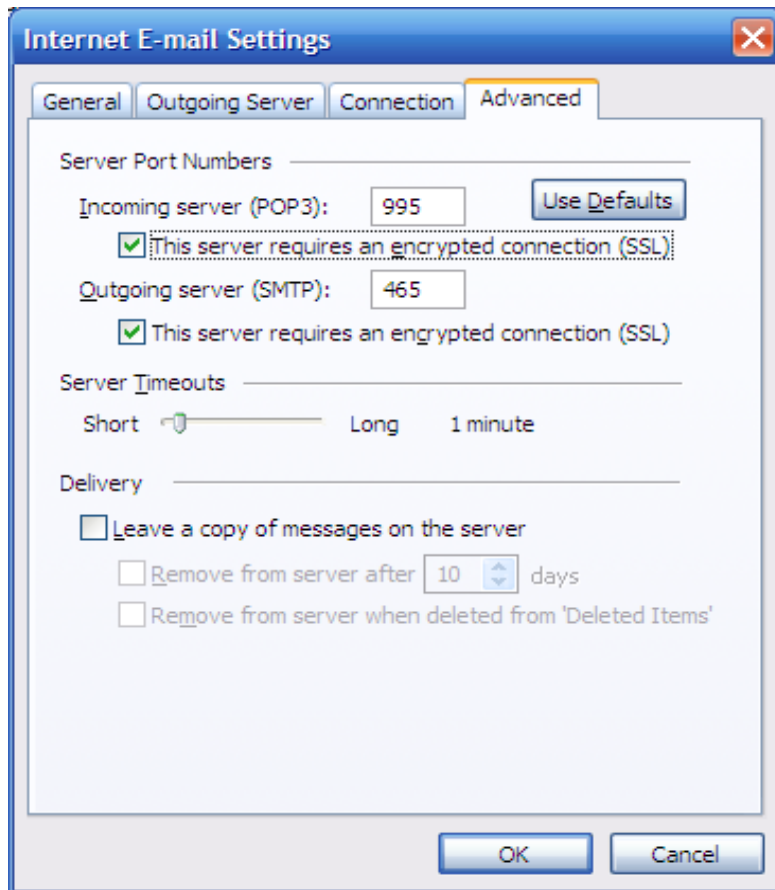
Điền '**pop.gmail.com**' vào ô '**Incoming mail (POP3) server:**'.

Điền '**smtp.gmail.com**' vào ô '**Outgoing mail (SMTP) server:**'.

( Incoming mail (POP3): Địa chỉ mail server mà mail client kết nối đến để lấy thư về .Outgoing mail (SMTP): Địa chỉ mail server mà mail client kết nối đến để gửi thư đi )

Điền Password vào ô '**Password:**'

Vào mục **More Settings..**



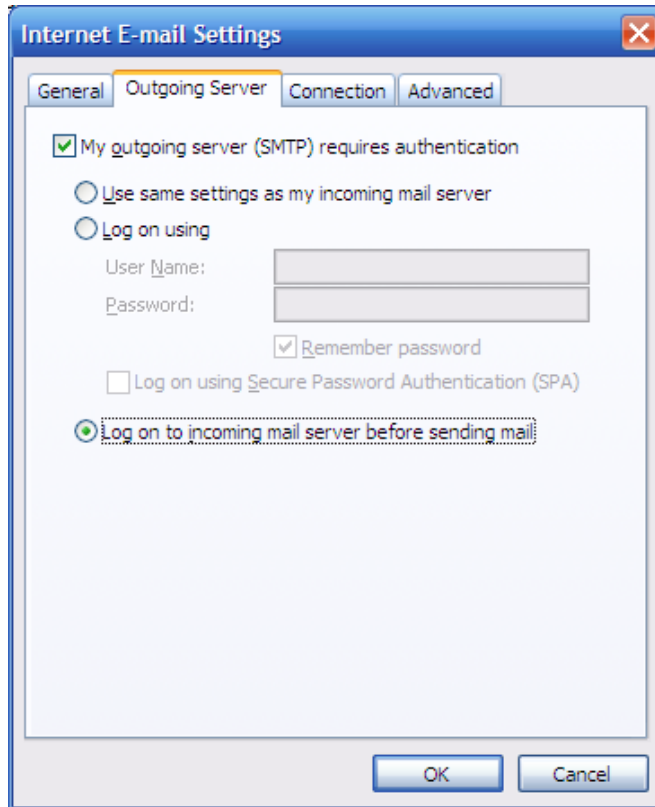
Click thẻ '**Advanced**'.

Đánh dấu chọn vào ô ở bên cạnh '**This server requires a secure connection (SSL)**' bên dưới '**Outgoing Mail (SMTP)**'.

Điền '**465**' vào ô '**Outgoing mail (SMTP)**'.

Đánh dấu chọn vào ô ở bên cạnh '**This server requires a secure connection (SSL)**' bên dưới '**Incoming mail (POP3)**'.

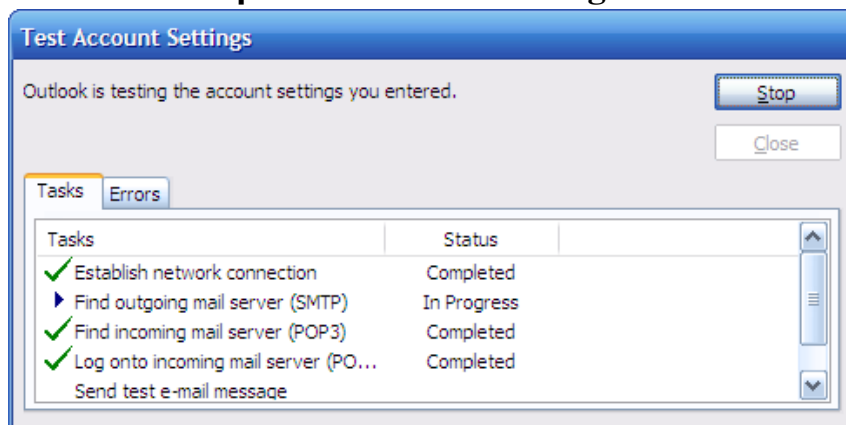
Cổng này sẽ chuyển sang 995.

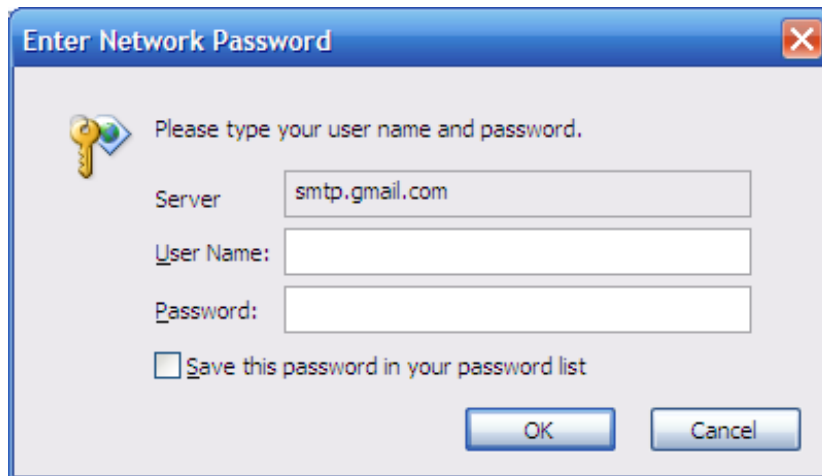


Chọn thẻ ' **Outgoing Servers**', và đánh dấu chọn vào ô bên cạnh '**My Outgoing server requires authentication.**'

Chọn Ok.

**Có thể vào mục Test Account settings để kiểm tra .**





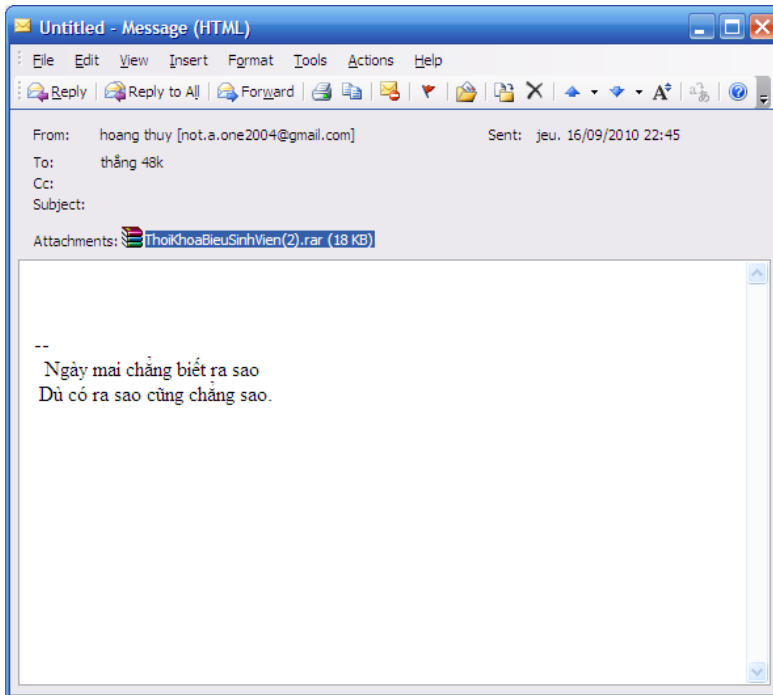
Kết thúc phần cài đặt, chọn Finish.

### 3.5. Hướng dẫn sử dụng Outlook Express

#### 3.5.1. Mở tệp đính kèm thư, ghi lại tệp đính kèm lên máy

Để mở một tệp tin đính kèm  
Bạn nhấn chuột vào bức thư có biểu tượng tệp tin đính kèm, biểu tượng đính kèm đó sẽ hiện ra to hơn (vùng đánh dấu). Nhấn chuột vào biểu tượng này, tên của tệp tin đính kèm sẽ hiện ra. Bạn tiếp tục nhấn vào tên tệp tin này.





Một cửa sổ hiện ra, bạn nhấp chuột phải chọn Open.

Để sao lưu tệp tin đính kèm vào máy tính

Cách 1:

Giống như cách 1 của mở tệp tin đính kèm, tuy nhiên khi có cửa sổ yêu cầu hiện ra, bạn nhấn chuột lựa chọn mục Save as...

Cách 2:

Nhấn chuột chọn bức thư có tệp tin đính kèm

Nhấn chuột vào menu **File** và chọn **Save Attachments**. Nếu thư có nhiều hơn một tệp tin đính kèm hộp thoại **Save Attachments** sẽ xuất hiện và cho phép bạn chọn những tệp tin muốn lưu lại.

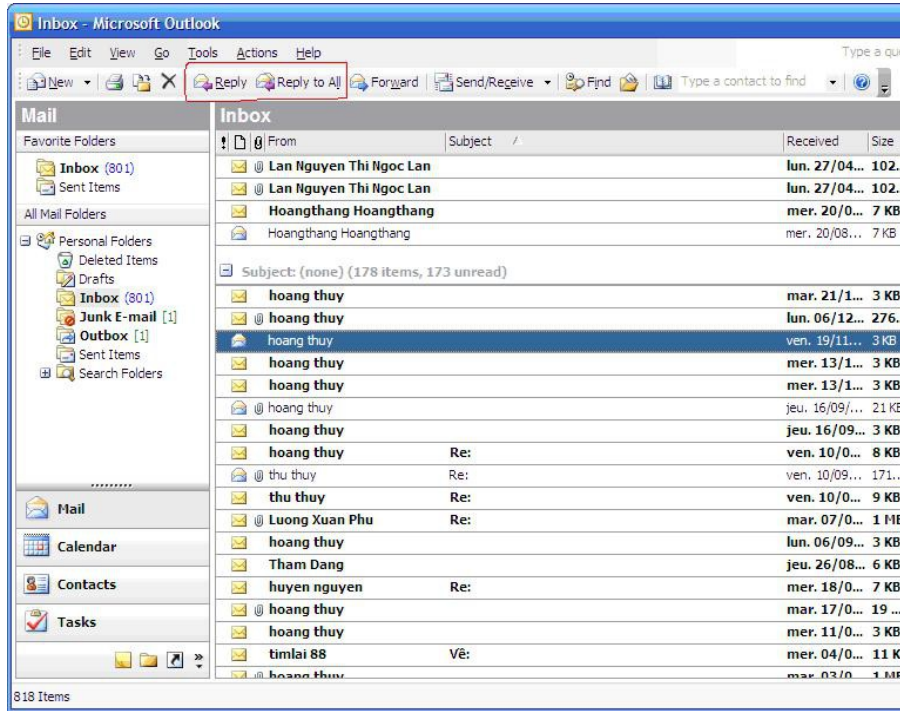
Chọn các tệp tin theo yêu cầu và nhấn chuột vào nút **OK**.

Hộp thoại **Save Attachment** hoặc **Save All Attachments** sẽ xuất hiện. Chọn thư mục muốn lưu và nhấn chuột vào nút **Save**.

Nhấn chuột vào nút **Close** để đóng hộp thoại Save Attachments.

### 3.5.2. Gửi thư trả lời

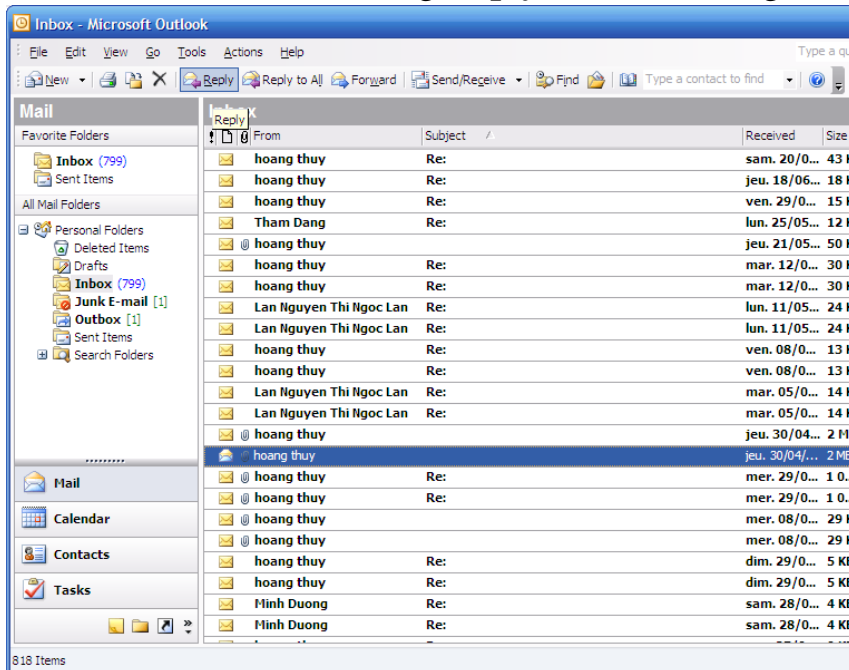
Dùng chức năng trả lời thư người gửi (**reply**) và trả lời tất cả (**reply to all**).



Để trả lời một bức thư :

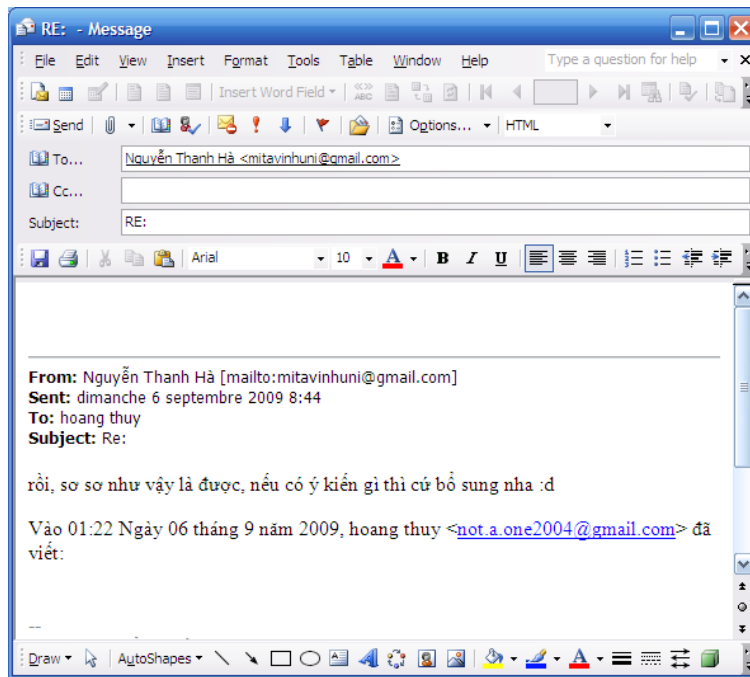
Trong thư mục **Inbox**, chọn bức thư mà bạn muốn trả lời .

Nhấn chuột vào biểu tượng **Reply** trên thanh công cụ .



Bạn sẽ thấy địa chỉ của người gửi thư cho bạn lúc này đã được tự động điền vào mục **To (Gửi tới)**, tiêu đề bức thư người đó gửi cho bạn cũng được tự

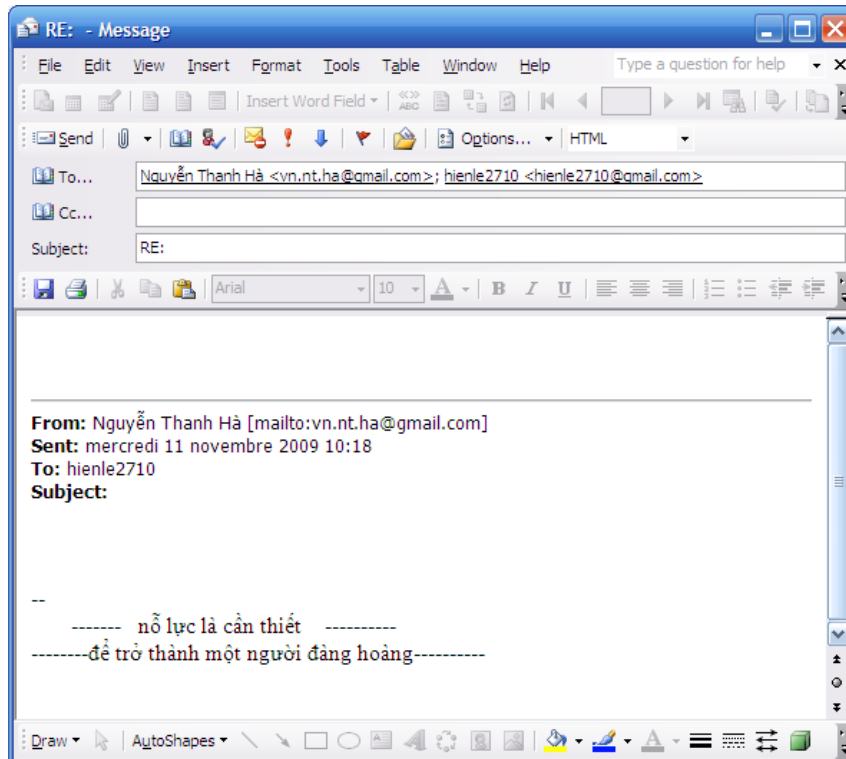
động điền vào mục **Subject (tiêu đề)**, nhưng trước đó có chữ Re:, có nghĩa là Trả lời cho nội dung ...(Re là viết tắt của chữ Reply - Trả lời).



Sau khi gõ nội dung trả lời, bạn có thể nhấn chuột vào nút **Send** trên thanh công cụ của hộp thoại Message để gửi thư đi.

Để trả lời người gửi và mọi người nhận khác:

Bạn nhận được một bức thư, không những gửi cho bạn mà còn gửi cho nhiều người khác nữa (ví dụ như một bức thư nhắc nhở cả nhóm cần hoàn thành công việc). Bạn muốn trả lời lại người gửi, đồng thời muốn cho cả những người mà đã được nhắc nhở biết thông tin trả lời của bạn, bạn nhấn vào nút **Reply to all**



### 3.5.3. Trả lời có kèm thêm thư gốc, không kèm thêm thư gốc

Trong chức năng trả lời thư, theo mặc định, nội dung lá thư gốc luôn hiện ra phía dưới cửa sổ soạn thư trả lời để giúp người trả lời có thể trích dẫn những thông tin mà người gửi hỏi để từ đó trả lời sát với nội dung thảo luận. Nội dung thư gốc thường đi sau chữ **---Original Message---**, hoặc chữ **Sender wrote that** (người gửi viết rằng) .

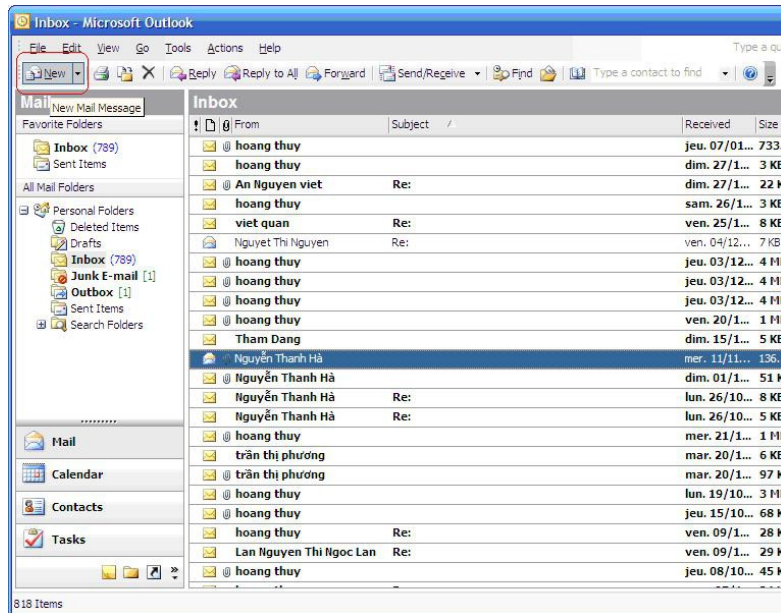
### 3.5.4. Gửi thư

#### 3.5.4.1. Tạo một bức thư mới :

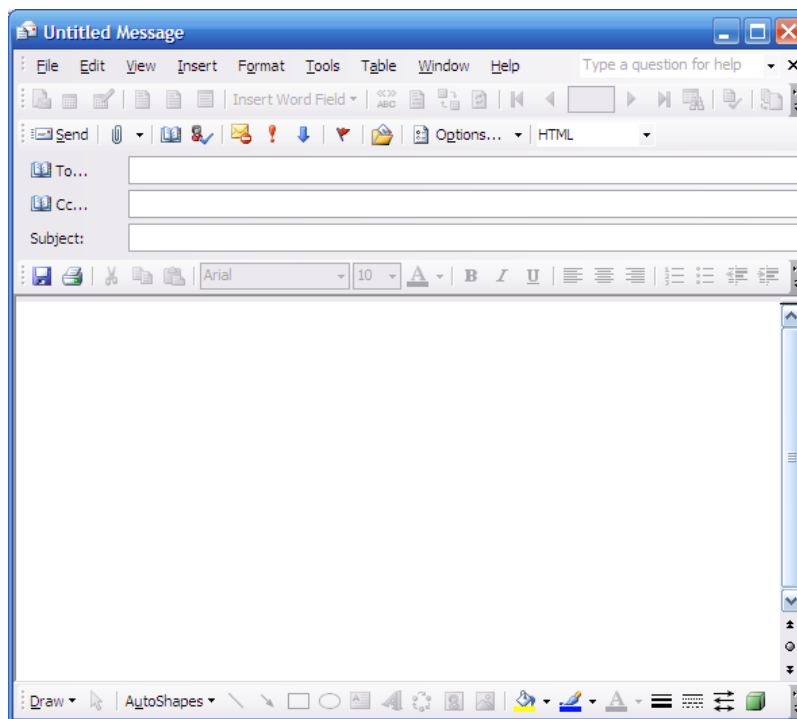
Khởi tạo một bức thư mới :

Để khởi tạo một bức thư mới

Chọn biểu tượng **Create Mail** trên thanh công cụ Hoặc nhấn **Ctrl + N** để hiển thị cửa sổ **New Message** cho phép soạn thư

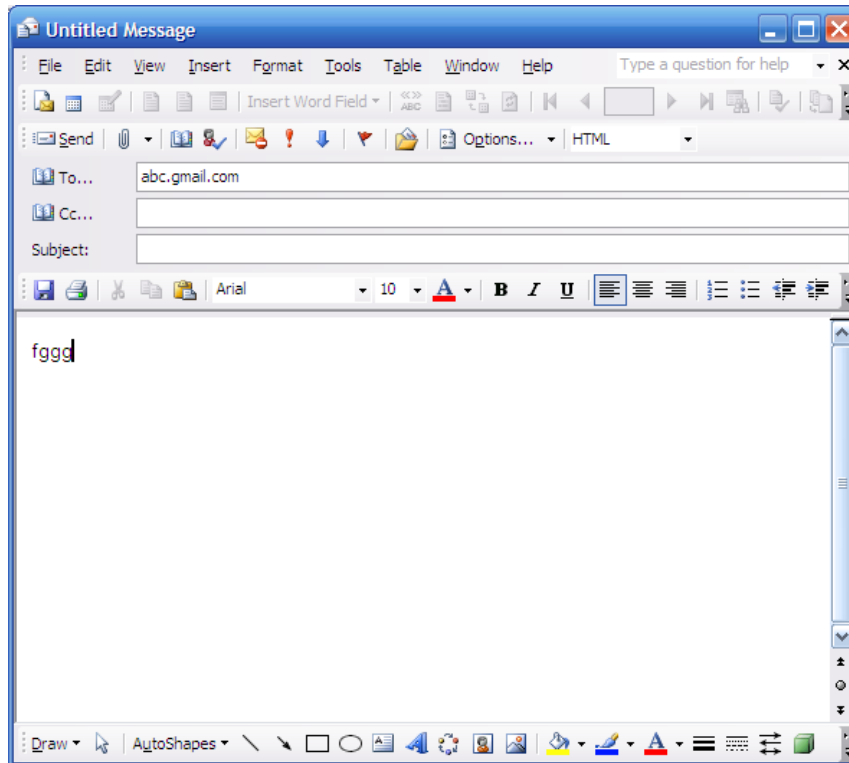


Một cửa sổ sẽ hiện ra, sẵn sàng đợi bạn nhập liệu các thông số cho bức thư mới



Điền một địa chỉ thư vào trường 'To...'

Để chỉ rõ người nhận thư là ai, gõ địa trực tiếp chỉ email của người nhận vào hộp địa chỉ To...:



Bạn cũng có thể lựa chọn địa chỉ bằng cách nhấn chuột vào biểu tượng quyển sổ ở cạnh nút To... để sử dụng chức năng chọn tên trong danh mục (Select Names).

Sử dụng hộp thoại **Select Recipients**, nhấn chuột vào nút **To** để mở hộp thoại Select Recipients .

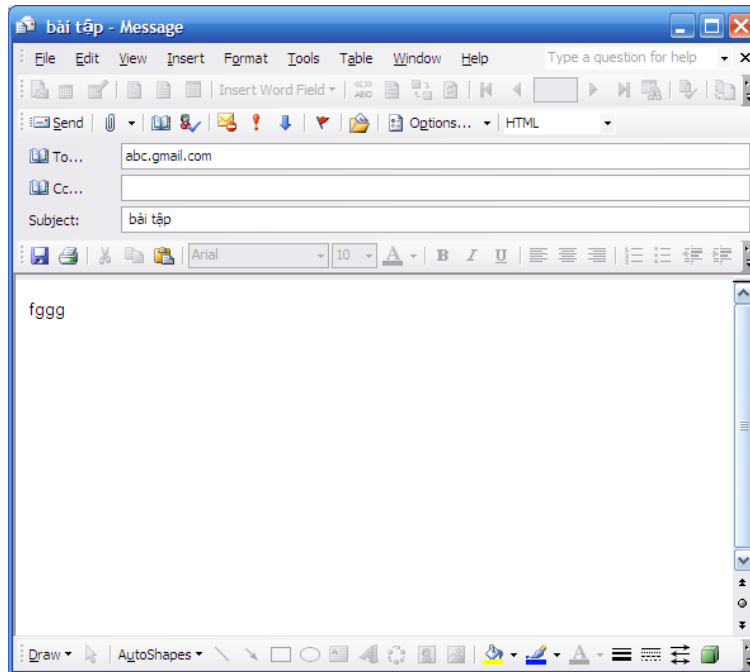
Chọn tên người nhận từ trong danh sách. Sau đó quyết định sẽ gửi thư cho những ai bằng cách chọn tên người đó và nhấn vào nút **To** -> để chuyển sang phần gửi tương ứng.

Tiếp tục lặp lại các thao tác như trên để có thể gửi thư cho nhiều người

Nhấn chuột vào nút **OK** để đóng hộp thoại Select Recipients và quay lại cửa sổ Message

Thêm tiêu đề vào trường chủ đề thư 'Subject'

Để đặt chủ đề (tiêu đề) cho bức thư bạn cần nhớ lại các quy tắc xã giao trên mạng: Viết tiêu đề thư ngắn gọn, súc tích, chỉ nêu ý chính cần trao đổi... và nội dung này được bạn gõ vào mục Subject

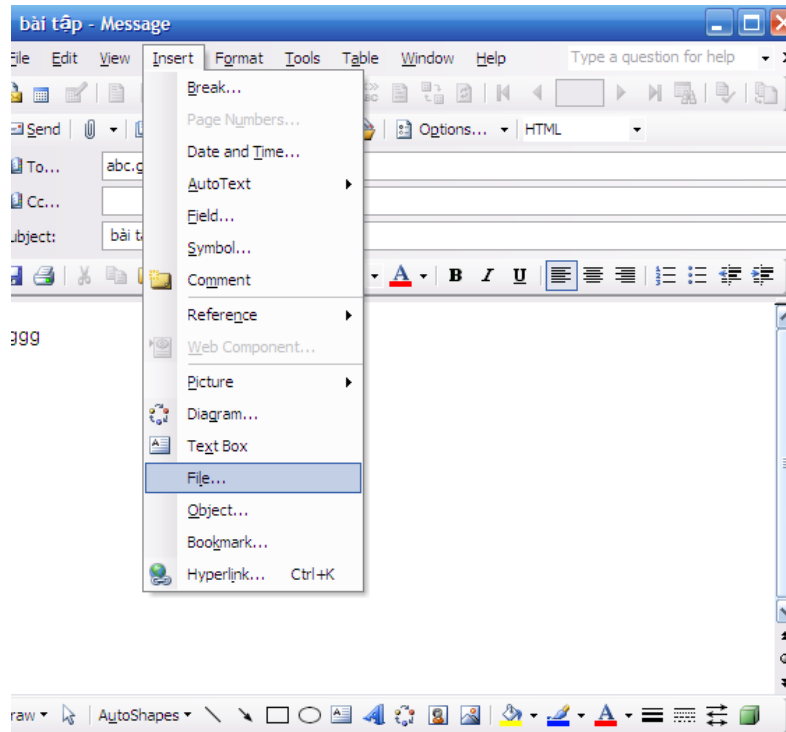


### 3.5.5. Đính kèm tệp tin vào thư

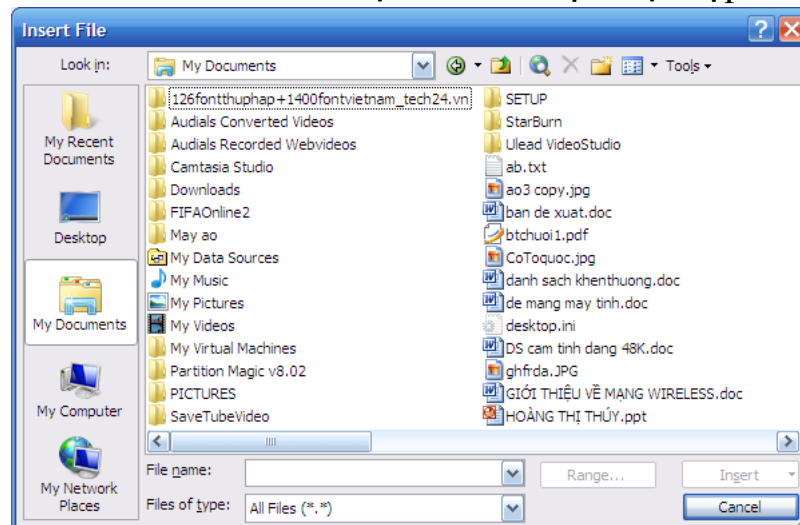
Để đính kèm một tệp tin vào bức thư đang soạn, ngoài nội dung thông tin gõ trong thư, đôi khi bạn còn muốn gửi thêm những tệp tin khác, như văn bản Word, bảng tính Excel, bài hát, đoạn phim... Việc đính kèm các tệp tin là không giới hạn về thể loại, còn số lượng các tệp tin được đính kèm thì tùy thuộc vào từng hệ thống thư tín điện tử.

Chú ý: nếu muốn đính kèm cả một thư mục, bạn cần nén (compress) thư mục đó lại thành một tệp tin thì mới có thể gửi đính kèm được.

Khi bạn đã gõ xong nội dung thư, muốn đính kèm theo tệp tin, bạn nhấn chuột vào biểu tượng Attach có hình kẹp ghim trên thanh công cụ, hoặc nhấn chuột vào menu Insert và chọn **File Attachment...**



Hộp thoại **Insert Attachment** hiện ra. Cần chọn tệp tin nào muốn đính kèm vào vào bức thư, bạn nhấn chuột chọn tệp tin đó rồi nhấn **Attach**

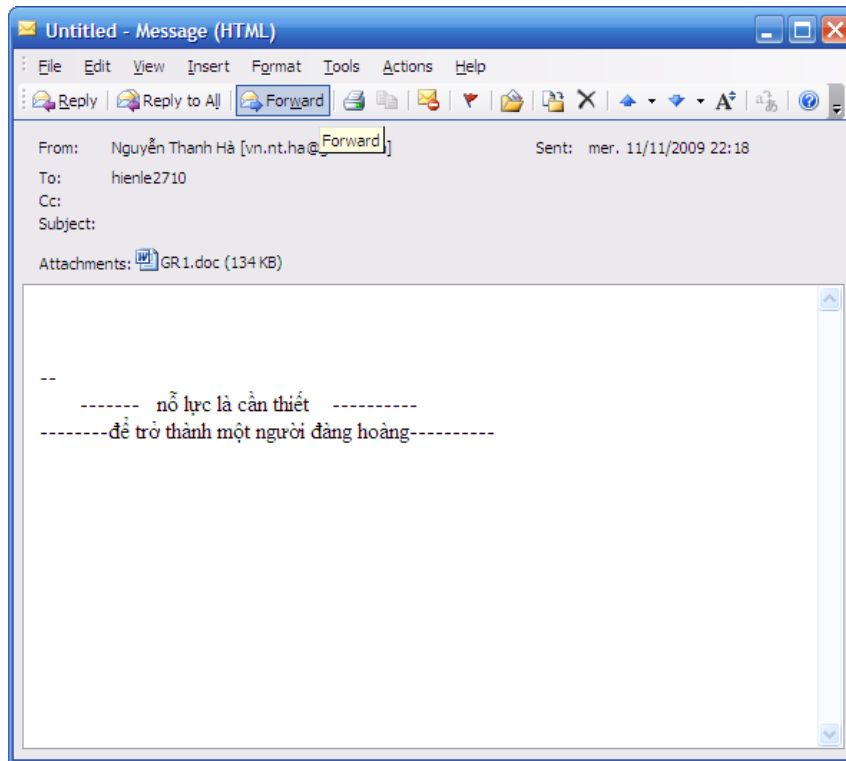


### 3.5.6. Forward (chuyển tiếp) một thư

Đôi khi, bạn nhận được một bức thư, sau đó bạn muốn người khác cũng đọc được bức thư này. Như vậy, bạn cần phải chuyển tiếp bức thư sang người đó.

Để chuyển tiếp một bức thư bạn chọn bức thư mà bạn muốn chuyển tiếp, nhấn chuột vào biểu tượng Forward trên thanh công cụ.

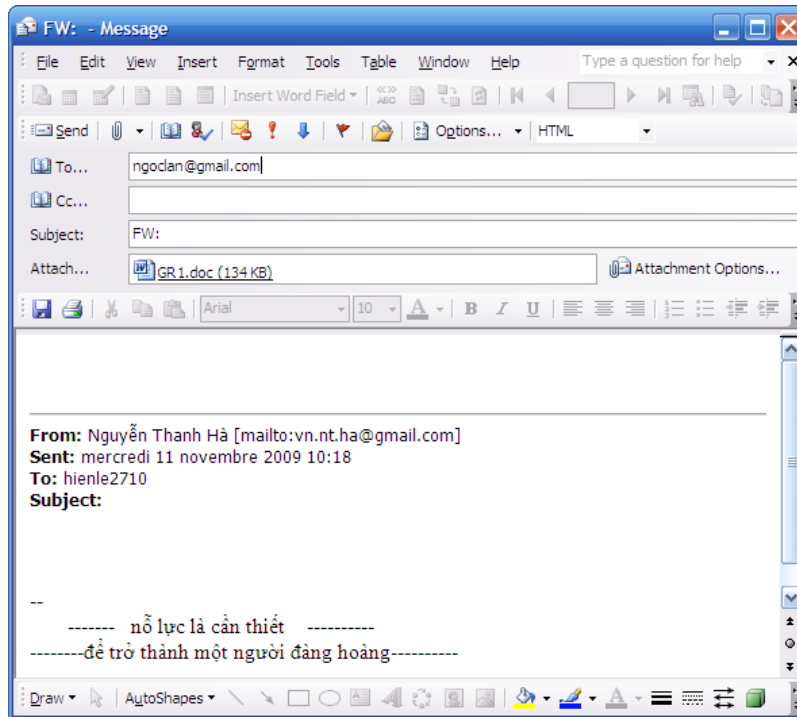




Hộp thoại chuyển tiếp thư sẽ hiện ra, bạn gõ địa chỉ của người mà bạn muốn chuyển tiếp vào mục To hoặc nhấn nút To và lựa chọn từ hộp thoại Select Names.

Bạn có thể gõ thêm vào nội dung thư cho người được chuyển tiếp biết những ý kiến của bạn khi chuyển lá thư này.

Sau đó, bạn nhấn Send để gửi thư đi



### 3.5.7. Sử dụng chức năng CC và BCC

#### a. CC, BCC là gì?

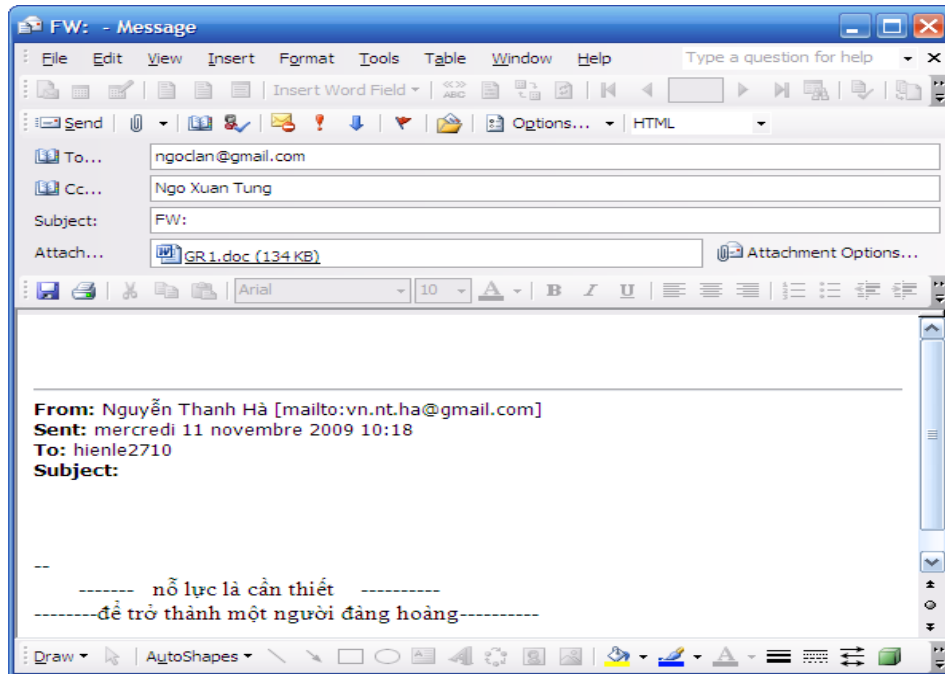
CC có nghĩa là Carbon Copy (bản sao theo lối giấy than) và BCC có nghĩa là Blind Carbon Copy (bản sao bí mật).

Cả hai chức năng CC và BCC đều cho phép bạn không những gửi thư cho người này mà còn đồng thời gửi nội dung đó cho nhiều người khác nữa.

Tùy theo tính chất công việc mà bạn chọn người nhận theo kiểu CC hay BCC. Nếu muốn công khai, bạn chọn CC, trong khi nếu không muốn cho người nhận biết mình còn gửi cho 1 người khác nữa, bạn chọn địa chỉ người được gửi bản sao theo kiểu BCC.

#### b. Gửi bản sao của thư theo hình thức CC :

Gõ vào mục **To** địa chỉ bạn muốn gửi chính, mục **CC** địa chỉ của người cũng sẽ nhận được nội dung bức thư này.



Trong ví dụ trên, bức thư sẽ được gửi tới cả hai địa chỉ và cả hai người này đều biết rằng ngoài mình ra, người kia cũng nhận được bức thư có nội dung tương tự.

## Câu 4: Từ Điển

### I. Giới Thiệu

Có một ứng dụng mà nó gắn liền với cuộc sống hằng ngày của chúng ta đó là

“ Từ Điển” , có thể bạn cần nó để tra cứu từ vựng, tra cứu danh mục hay giúp bạn tìm kiếm sách .v.v... Từ điển là ứng dụng có ưu thế về tốc độ, số lượng ứng dụng và cả về chất lượng của các từ điển, do nhu cầu mà ứng dụng từ điển đã phát triển rất đa dạng và phong phú. Tùy từng ứng dụng mà người dùng có thể tra cứu mọi thứ dễ dàng.

Cũng giống như mọi ứng dụng từ điển khác, bài từ điển này cũng có ý nghĩa và ứng dụng tương tự nhưng nó ở phạm vi nhỏ hơn đó là tra cứu từ trên một văn bản.

Để tạo được ứng dụng từ điển này ta cần phải tìm hiểu một số khái niệm như:

Danh sách là một dãy hữu hạn các phần tử thuộc một lớp đối tượng nào đó. Ví dụ: danh sách sinh viên, danh sách hàng, danh sách các số thực....

Để biểu diễn danh sách ta có thể dùng hai cách là dùng mảng và dùng danh sách liên kết.

Có thể hiểu đơn giản về mảng là tập hợp các phần tử kế tiếp nhau và hoàn toàn có vị trí cố định ( bộ nhớ tĩnh) , có thể đi đến trực tiếp.

Còn về danh sách liên kết thì áp dụng bộ nhớ động có nghĩa là muốn đi đến phần tử chúng ta cần thì phải bắt đầu từ phần tử đầu tiên.

Mảng và danh sách liên kết đều có ưu và nhược điểm

#### \*Mảng:

Dùng để lưu trữ danh sách các phần tử (lưu liên kế nhau) cùng kiểu.

#### Ưu điểm:

- Dễ cài đặt và truy nhập các phần tử dữ liệu
- Tốc độ truy nhập tới một vị trí bất kỳ trên mảng nhanh, và hiệu quả

Nhược điểm:

- cần phải xác định trước số phần tử mảng trước khi sử dụng ( kể cả mảng động) → không phù hợp với các bài toán chưa biết trước số lượng các phần tử .
- Khó khăn trong các thao tác chèn và xóa một phần tử bất kỳ trong mảng. Nếu bài toán mà việc chèn phần tử và xóa phần tử diễn ra liên tục thì tốc độ xử lý sẽ rất chậm.

**\*\*Danh sách liên kết:**

Cũng như mảng danh sách liên kết (DSLK) dùng để lưu trữ các phần tử. Tuy nhiên cách lưu trữ của DSLK khác hẳn với mảng. Với danh sách liên kết, các phần tử ngoài dữ liệu, nó còn lưu trữ các địa chỉ mà các phần tử có quan hệ với nó ( nhờ các địa chỉ này mà ta có thể tìm các phần tử khác nhau trong danh sách).

DSLK có rất nhiều loại nhưng cơ bản có hai loại : Danh sách đơn liên kết và Danh sách đa liên kết.

Ưu điểm:

- Do DSLK dùng tới đâu thì cấp phát tới đó → phù hợp với các bài toán mà các phần tử chưa xác định trước.
- Kích thước danh sách chỉ bị giới hạn do kích thước RAM.
- Dễ dàng trong việc xóa và chèn các phần tử trong danh sách (chỉ cần thay đổi các địa chỉ của các phần tử có quan hệ với nó cho phù hợp là được )

Nhược điểm:

- Khó cài đặt và truy nhập đến các phần tử trong danh sách
- Tốc độ truy nhập của danh sách chậm → do đó người ta thường dùng các cây quyết định để phục vụ cho việc tìm kiếm trong danh sách ( cây nhị phân tìm kiếm ...)

Vì bài toán ứng dụng cho một tệp văn bản mà chưa xác định được độ dài, và cần chèn, xóa các từ trong văn bản . Vì thế ta dùng danh sách liên kết đơn để biểu diễn danh sách trong ứng dụng này là hợp lý nhất.

Danh sách liên kết có thể được thực hiện trong hầu hết các ngôn ngữ. Như ngôn ngữ pascal, ngôn ngữ hướng đối tượng, java, v.v.

Lựa chọn ngôn ngữ Turbo C là ngôn ngữ thông dụng hiện nay để thực hiện bài toán lưu trữ và sắp xếp từ điển, một phần nữa là để ôn lại các kiến thức đã học trong các môn như “Cấu trúc dữ liệu và giải thuật”, “Ngôn ngữ lập trình turbo C..”

## II. Báo Cáo.

### 1. Bài toán Từ Điển:

Cho tập văn bản trong đó chứa các từ, các dấu phân cách từ: dấu cách, dấu phẩy, dấu chấm phẩy, dấu hai chấm, dấu than, dấu hỏi. Mọi từ bắt đầu bằng chữ cái 'A'..'Z' (không phân biệt chữ hoa và chữ thường).

1. Viết thủ tục đọc các từ trong tập văn bản và lưu trữ vào mảng các danh sách liên kết:

```
Type DanhSach=^PhanTu;
PhanTu=Record
Tu:String[15];
Tiep:DanhSach;
End;
```

Var

```
TuDien: array['A'..'Z'] of DanhSach;
```

Mỗi danh sách liên kết trong từ điển xếp theo thứ tự tăng và các từ khoá là khác nhau.

2. Viết thủ tục hiển thị các từ ra màn hình theo thứ tự tăng.

3. Viết thủ tục bổ sung từ mới bằng cách đọc từ bàn phím và tìm nó trong từ điển.

- Nếu thấy, thì hiển thị thông báo 'Đã có'.

- Nếu không, chèn vào vị trí thích hợp.

4. Viết Menu để thực hiện các chức năng trên.

### 2. Nguyên tắc tổ chức danh sách đơn.

Mỗi phần tử của danh sách được lưu trữ trong một phần tử nhớ mà ta gọi là nút(node). Các nút này có thể nằm ngoài bất kỳ ở chỗ nào trong bộ nhớ. Trong mỗi nút ngoài phần thông tin ứng với một phần tử còn chứa địa chỉ của các phần tử đứng sau nó trong danh sách.

Mỗi nút có thể hình dung như sau:

DATA	LINK
------	------

Trong đó: Trường

DATA : chứa thông tin của phần tử(của nút).

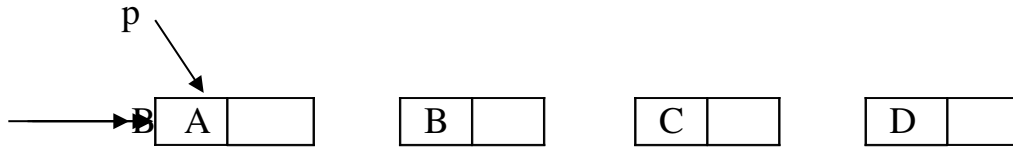
Trường LINK : chứa địa chỉ(mối nối) của nút kết tiếp trong danh sách.

Riêng nút cuối cùng thì không có nút đứng sau nó nên mối nối ở nút này phải là một "địa chỉ đặc biệt" chỉ dùng để đánh dấu nút kết thúc danh sách

chứ không như các địa chỉ ở các nút khác ta gọi là “mối nối không” và ký hiệu là NULL.

Để có thể truy nhập đến các nút của danh sách cần thiết phải sử dụng thêm một biến trỏ trỏ đến nút đầu tiên, giả sử là P, và quy ước danh sách rỗng (P=NULL).

Minh họa danh sách:



Dấu nhạn chỉ mối nối không hay là NULL  
A,B tượng trưng cho phần biểu diễn thông tin .

Khai báo kiểu điều kiện:

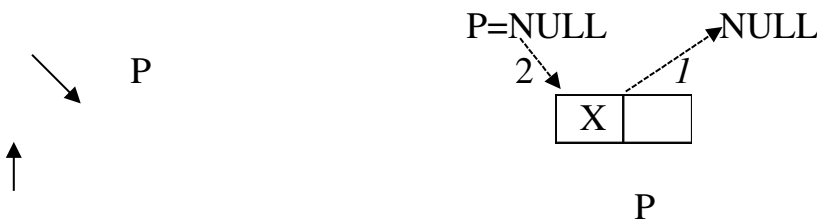
```
Struct node
{ Char data[15];
  Struct node*link;
};
```

### 3.Một số phép toán:

a/Bổ sung một nút mới vào danh sách.

Bài toán: Lập giải thuật bổ sung một nút mới có Data = st1 vào sau nút trỏ bởi h(nếu danh sách khác rỗng)của danh sách liên kết đơn trỏ bởi P:

\*Trường hợp Danh sách rỗng:  $P=NULL$ .

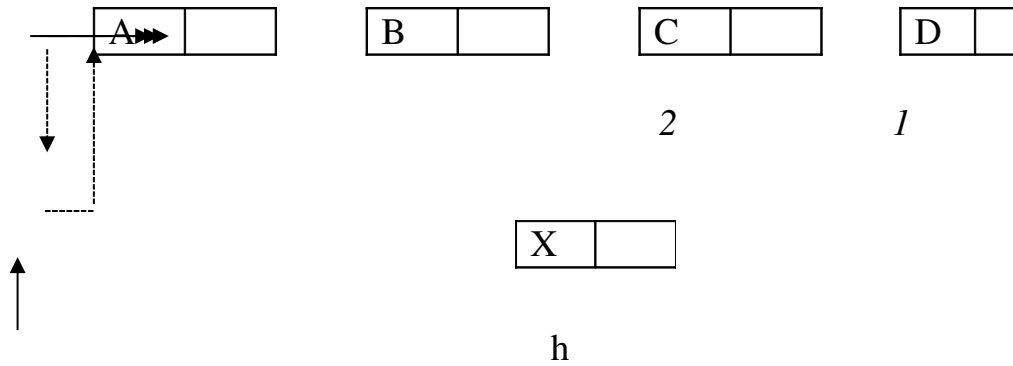


-Nét liền là những cái đã có(cho trước).

-Nét đứt và số thứ tự là công việc cần làm.

\*Trường hợp danh sách khác rỗng:  $P \neq NULL$ .





**\*\*Giải thuật bổ sung một nút mới vào danh sách:**

```
void bosung(char *stl)           //thu tuc dua mot tu vao danh sach tu
dien
```

```
{
    node *q,*h,*p;
    node *t[100];
    char ch;
    q=t[ch];
    h=q;
    p=new node;
    strcpy(p->data,stl);
    p->link=NULL;
    if(t[ch]==NULL)           //Truong hop danh sach rong
    {
        t[ch]=p;
    }else
    {
        if(strcmp(t[ch]->data,stl)>0)
        {
            p->link=t[ch];
            t[ch]=p;
        }else
        {
            while((q!=NULL)&&(strcmp(stl,q->data)>0))
            {
                h=q;
                q=q->link;
            }
            h->link=p;
        }
    }
}
```



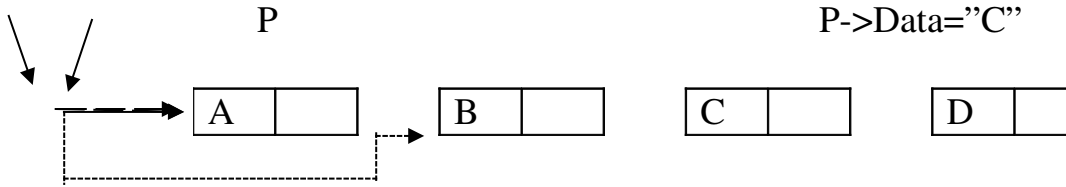
```

        p->link=q;
    }
}

```

***b/Loại bỏ một nút ra khỏi danh sách:***

**Bài Toán:** Loại bỏ nút có dữ liệu nhập vào (data=st)(nếu danh sách P≠NULL) ra khỏi danh sách liên kết đơn.



/

**\*\*Giải thuật loại bỏ nút có Data= st:**

```

int xoatu(char *st)
{
    node *p,*q;
    p=t[toupper(st[0])];
    q=p;

    /*truong hop loai trong danh sach khac rong*/
    /*tim q nut lien truoc p)*/
    while((p!=NULL)&&(strcmp(p->data,st)!=0))
    {
        q=p;
        p=p->link;
    }
    if(p==NULL) /*truong hop danh sach rong
*/
    {
        return 0;
    }
    Else /*truong hop loai nut dau
tien*/
    {

```

```

        q->link=p->link;
        if(p==t[toupper(st[0])])
        {
            t[toupper(st[0])]=p->link;
        }
        delete(p);
        return 1;
    }
}

```

### c/Tìm kiếm một nút có trong danh sách không:

Bài toán: Lập hàm tìm nút có dữ liệu data=st trong danh sách nếu có thì trả về giá trị 1 ngược lại trả về giá trị 0:

//ham nay kiem tra xem mot tu da co trong danh sach chua !

```

int ktxaudaco(char *st)
{
    node *p;
    p=t[toupper(st[0])];
    while((p!=NULL)&&(strcmp(p->data,st)!=0))
    {
        p=p->link;
    }
    if(p==NULL)
        return 0;
    else
        return 1;
}

```

### III.Các modul chính trong chương trình .

#### 1.Đọc nội dung của văn bản và đưa vào từ điển:

Đọc tệp và lưu vào từ điển, loại bỏ các dấu ( dấu cách, dấu phẩy, dấu chấm,dấu chấm phẩy, dấu hai chấm, dấu chấm than, dấu hỏi), tách từ và gọi hàm

*Capnhattutiep(tentep)* để lưu vào danh sách liên kết.

## **2.Hiển thị từ điển ra màn hình.**

Hiển thị danh sách liên kết ra màn hình theo thứ tự tăng dần.

## **3.Tìm kiếm từ.**

Sử dụng thuật toán tìm kiếm, tìm một từ trong danh sách.Nếu có in ra thông báo “từ này đã có trong từ điển”, ngược lại nếu chưa có trong từ điển thì bổ xung từ đó vào trong từ điển và hiện thông báo “cập nhật thành công”.

## **4.Loại bỏ từ.**

Nhập vào từ cần loại bỏ.Nếu có trong danh sách thì xoá từ và in ra màn hình thông báo đã loại bỏ,ngược lại in ra màn hình thông báo từ này không có trong từ điển.

## **5.Ghi từ điển ra tệp.**

Nhập tên tệp cần in từ điển, in ra màn hình thông báo: ghi tệp thành công.

## **6.Thoát chương trình.**

# **IV.Chương trình:**

## **1.Chương trình nguồn:**

```
#include<conio.h>
#include<stdio.h>
#include<stdlib.h>
#include<ctype.h>
#include<string.h>
struct node
{
    char data[15];
    struct node *link;
};
//khai bao bien
```

```

node *t[100];
char st[100],tu[100],tentep[30];
FILE *f;
char ch;
int so;
//ham ktra ki tu co phai la cac dau .,?!:; ko.neu co thi tra ve vi tri trong xau
int poss(char *st)
{
    int luu;
    int i, len,j;
    int ktra;
    fflush(stdin);
    char *xau=".,!?:;";
    ktra=1;
    len=strlen(st);
    luu=len;
    for(i=0; i<len; i++)
        {
            for(j=0; j<7; j++)
                {
                    if(st[i]==xau[j])
                        {
                            luu=i;
                            ktra=0;
                            break;
                        }
                }
            if(!ktra) break;
        }
    return luu;
}
//ham ktra xem mot tu da co trong danh sach chua

int ktxaudaco(char *st)
{
    node *p;

```

```

p=t[toupper(st[0])];
while((p!=NULL) &&(strcmp(p->data,st)!=0))
{
    p=p->link;
}
if(p==NULL)
{
    return 0;
}
else
{
    return 1;
}
}
/* ham kiem tra xem mot tu da co trong ds chua neu chua thi thong
bao ko co*/
//neu co thi xoa

int xoatu(char *st)
{
    node *p,*q;
    p=t[toupper(st[0])];
    q=p;
    while((p!=NULL) &&(strcmp(p->data,st)!=0))
    {
        q=p;
        p=p->link;
    }
    if(p==NULL)
    {
        return 0;
    }
    else
    {
        q->link=p->link;
        if(p==t[toupper(st[0])])
        {

```

```

        t[toupper(st[0])]=p->link;
    }
    delete(p);
    return 1;
}
}
//thu tục đưa một từ vào danh sách từ điển
void bosung(char *st1)
{
    node *q,*h,*p;
    q=t[ch];
    h=q;
    p=new node;
    strcpy(p->data,st1);
    p->link =NULL;
    if(t[ch]==NULL)
    {
        t[ch]=p;
    }
    else
    {
        if(strcmp(t[ch]->data,st1)>0)
        {
            p->link=t[ch];
            t[ch]=p;
        }
        else
        {
            while ((q!=NULL)&&(strcmp(st1,q->data)>0))
            {
                h=q;
                q=q->link;
            }
            h->link =p;
            p->link=q;
        }
    }
}
}

```

```

    }
    //tach cac tu(loai cac dau ,.!?;) trong mot xau dung ham poss de lay //vi
    tri cua cac dau trong xau
    void tachtu(char *st)
    {
        int len;
        char *st1;
        int vt;
        len=strlen(st);
        while(len>0)
        {
            vt=poss(st);
            for(int i=0; i<vt; i++)
                st1[i]=st[i];
            st1[i]='\0';

            int j=0;
            for(i=vt+1; i<len; i++)
                st[j++]=st[i];
            st[j]='\0';

            if ((st1!=NULL)&&strcmp(st1," "))
            {
                strcpy(st1,strupr(st1));
                ch=st1[0];
                if(!ktxaudaco(st1))
                {
                    bosung(st1);
                    printf("\n %s\n", st1);
                }
            }
            len=strlen(st);
        }
    }
    //thu tuc in danh sach liet ke ra man hinh
    void inds(node *l)
    {

```

```

node *p ;
p=l;
while (p!=NULL)
{
    printf("\n %s", p->data);
    p=p->link;
}
printf("\n\n");
}
//thu tục ghi danh sách vào tệp
void ghifile(node *l)
{
    node *p;
    p=l ;
    while(p!=NULL)
    {
        fprintf(f,"%s",p->data);
        p=p->link;
    }
    fprintf(f,"\n");
}
//in tu dien ra màn hình

void intudian()
{
    for(int ch='A';ch<='Z'; ch++)
    {
        if(t[ch]!=NULL)
        {
            inds(t[ch]);
            getch();
        }
    }
}
//thu tục đọc các từ trong tệp và đưa nó vào từ điển
void capnhattutep(char *g)
{

```



```

f=fopen(g,"rt");
while(!feof(f))
{
    fscanf(f,"%s" ,st) ;
    tachtu(st);
}
fcloseall();
}
//thu tục ghi tu dien vao tep
void ghitep(char *g)
{
    f=fopen(g,"wt");
    for(ch='A'; ch<'Z'; ch++)
    {
        if(t[ch]!=NULL)
        {
            ghifile(t[ch]);
        }
    }
    fcloseall();
}
//menu chương trình
void menu()
{
    clrscr();
    printf("\n\n");
    printf("\n      Moi Ban Chon Ung Dung:\n\n ");
    printf("\t1: Doc noi dung cua van ban va dua vao tu dien: \n" );
    printf("\t2: Hien thi tu dien ra man hinh.\n" );
    printf("\t3: Tim kiem tu.\n");
    printf("\t4: Loai bo tu.\n");
    printf("\t5: Ghi tu dien ra file.\n");
    printf("\t6: Thoat chương trình.\n");
    printf("\n");
    printf(" Nhap ung dung ban chon la so :" );
    scanf("%d",&so);
}

```

```
// than chuong trinh
void main()
{ clrscr();
  so =0;
  while (so!=6)
  { menu();
    printf("%d",so);
    switch(so)
    {
      case 1:
        clrscr();
        printf("Nhap ten tep can nhap vao tu dien: ");
        strcpy(tentep," ");
        while(!strcmp(tentep," "))
        {
          fflush(stdin);
          gets(tentep);
        }
        capnhattutep(tentep);
        printf("Cap nhat thanh cong \n ");
        getch();
        break;

      case 2:
        clrscr();
        printf("danh sach tu trong tu dien : \n");
        intudien();
        getch(); break;

      case 3:
        clrscr();
        printf("\n");
        printf("Moi ban nhap tu can tim : ");
        strcpy(tu," ");
        while(!strcmp(tu," "))
        {
          fflush(stdin);
          gets(tu);
        }
    }
  }
}
```

```

}
strcpy(tu,strupr(tu));
if(ktxaudaco(tu))
{
    printf("Tu nay da co trong tu dien !\n" );
    getch();
}
else
{ printf("Tu nay chua co trong tu dien " );

ch=tu[0];
bosung(tu);
printf("Cap nhat tu moi thanh cong !\n");
getch();
}
strcpy(tentep," ");
break;

```

case 4:

```

clrscr();
printf("\n");
printf("Moi ban nhap tu can loai bo:");
strcpy(tu, " ");
while(!strcmp(tu, " "))
{
    fflush (stdin);
    gets(tu);
}
strcpy(tu,strupr(tu));
if(xoatu(tu))
{
    printf("Da loai bo !\n");
    getch();
}
else
{
    printf(" Tu nay khong co trong tu dien .\n" );

```

```

        getch();
    }
    strcpy(tentep, " ");
    break;
case 5:
    clrscr();
    printf("\n");
    printf("Moi ban nhap ten tep can back up :");
    strcpy(tentep, "");
    while(!strcmp(tentep, ""))
    {
        gets(tentep);
    }
    ghitep(tentep);
    printf("\n Ghi file thanh cong !");
    getch();
    break;
case 6:
    exit(1);
}
}
getch();
}

```

## 2.Thực hiện chương trình:

Để chạy chương trình trước hết tạo một văn bản trên Notepad. Ví dụ bạn có văn bản như sau:

Bài thơ: Con Co !!

Cai co di don con mua  
Toi tam mu mit ai đưa co ve  
Co ve tham quan cung que  
Tham cha tham me, co ve tham em

Sau đó lưu với tên: **conco.txt** (Lưu vào thư mục Bin của chương trình C )

Chạy chương trình. **Ctrl +F9**.

Tại giao diện đầu tiên: Chọn ứng dụng mà bạn muốn thực hiện.

**Nhập số 1:**(Chọn ứng dụng đọc nội dung văn bản và đưa và từ điển)

**Enter:**

Lúc này bạn nhập tên văn bản mà bạn cần đưa vào : **conco.txt /Enter**

Màn hình chỉ xuất hiện các từ trong văn bản mà đã loại các kí tự dấu chấm, dấu phẩy, dấu chấm than...và các từ giống nhau chỉ xuất hiện 1 lần.

*Nhap ten tep can nhap vao tu dien: conco.txt*

*Bai  
Tho  
Con  
Co  
Cai  
Di  
Don  
Mua  
Toi  
Tam  
Mu  
Mit  
Ai  
Dua  
Ve  
Tham  
Quan  
Cung  
Que  
Cha  
Me  
Em*

Vì từ điển chứa đựng một tập hợp các đơn vị ngôn ngữ, từ vựng, sắp xếp theo thứ tự dễ tra, dễ tìm, sắp xếp theo thứ tự chữ cái.

Nên khi bạn **nhập số 2** (ứng dụng in từ điển). Bạn sẽ nhận được một danh sách các từ trong văn bản đã được sắp xếp tăng dần theo bảng chữ cái.

Khi nhận được danh sách bằng cách **Enter** liên tục đến hết các từ.

*Danh sách từ trong từ điển:*

*Ai*

*Bai*

*Cai*

*Cha*

*Co*

*Con*

*Cung*

*Di*

*Don*

*Dua*

*Em*

*Me*

*Mit*

*Mu*

*Mua*

*Quan*

*Que*

*Tam*

*Tham*

*Tho*

*Toi*

*Ve*

Bạn ra màn hình chọn ứng dụng:

**Nhập số 3:**(Tìm kiếm từ)

Đây là chức năng chính của từ điển: bạn nhập từ cần tìm kiếm.

Nếu có trong văn bản conco.txt thì màn hình sẽ hiện thông báo “Từ này đã có trong văn bản”. Nếu từ bạn nhập vào không có thì chương trình sẽ bổ sung vào từ điển và hiện thông báo “Từ này không có trong từ điển”, “Cập nhật từ mới thành công”.

*Moi ban nhap tu can tim : quy*

*Tu nay khong co trong tu dien cap nhat tu moi thanh cong!*

Còn nếu bạn muốn xóa một từ nào đó trong từ điển bạn **nhập số 4** ở màn hình giao diện chính của chương trình rồi nhấn phím **Enter**. Màn hình mời bạn nhập từ cần xóa:

Chương trình sẽ kiểm tra; Nếu có chương trình sẽ xóa và hiện thông báo xóa thành công. Ngược lại thông báo là không tìm thấy từ trong văn bản.

Nhấn **Enter** bạn thoát ứng dụng ra màn hình chính:

*Moi ban nhap tu can loai bo: nong*

*Da loai bo!*

Bạn **nhập số 5**: Nếu bạn muốn lưu lại tất cả những thay đổi khi thực hiện các ứng dụng trên vào văn bản( có thể tạo một tệp văn bản khác để cập nhật). Bạn nhập tên của tệp văn bản ví dụ: **conco2.txt /Enter**

Tệp văn bản lúc này được cập nhật từ danh sách các từ trong từ điển(các từ viết hoa, được sắp xếp theo bảng chữ cái, không có các dấu chấm, dấu hỏi, dấu chấm than,.....).

Tùy sự lựa chọn của bạn mà bạn muốn thực hiện các ứng dụng. Nếu muốn thoát chương trình bạn ***nhấn số 6 / Enter***.