

NGÔN NGỮ LẬP TRÌNH

PASCAL

BÀI

1

GỚI THIỆU CHƯƠNG TRÌNH PASCAL

KIẾN THỨC YÊU CẦU

- Hiểu được các khái niệm cơ bản về ngôn ngữ Pascal.
- Biết cách cài đặt Pascal từ đĩa CD.
- Biết cách sử dụng Pascal 7.0.
- Biết tạo, lưu và mở tập tin chương trình.
- Biết biên dịch và thực thi chương trình.

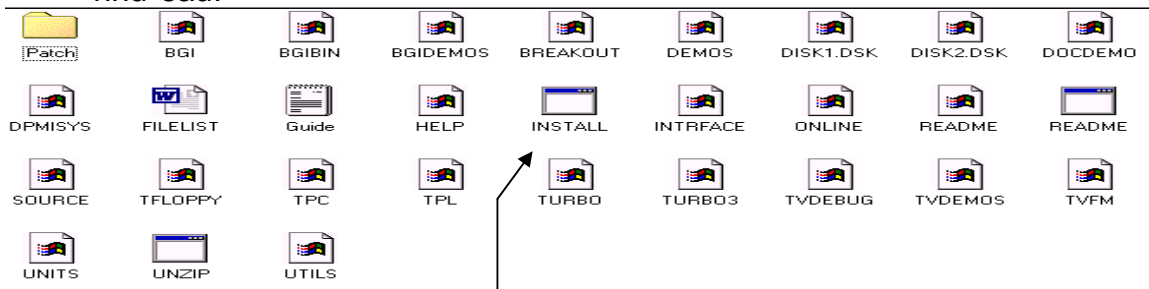
I/ Khái niệm

- **Ngôn ngữ lập trình** (programming language): Là một hệ thống các *kí hiệu* tuân theo các quy ước về *ngữ pháp* và *ngữ nghĩa*, dùng để xây dựng các chương trình cho máy tính.
 - **Ngữ pháp** (syntax): Quy ước về quan hệ giữa các ký hiệu. Ví dụ trong ngôn ngữ Pascal: các ký hiệu **Begin**, **end** phải đi thành từng cặp, sau **if** sẽ là một biểu thức điều kiện, sau đó là ký hiệu **then**.
 - **Ngữ nghĩa** (semantics): Quy ước về ý nghĩa của ký hiệu. Ví dụ trong ngôn ngữ Pascal: dấu + biểu thị cho phép cộng, dấu - biểu thị cho dấu trừ ...Phát biểu **if ...then ...** có nghĩa là “nếu ... thì làm ...”.
- **Chương trình** (program): Là một tập hợp các mô tả, các phát biểu, nằm trong một hệ thống quy ước về ý nghĩa và thứ tự thực hiện, nhằm điều khiển máy tính làm việc.
- **Lập trình** (to program): Làm ra chương trình, viết ra chương trình, còn gọi là “thảo chương”.

II/ Cài đặt chương trình

1/ Cài đặt chương trình từ đĩa CD chứa chương trình Turbo Pascal

- ✎ Cho đĩa CD có chứa Turbo Pascal 7.0 (TP7.0) vào ổ đĩa CD.
- ✎ Vào cửa sổ **My Computer** hay **Explore**.
- ✎ Nhấp phải chuột trên hình đĩa CD, chọn **Open** trong menu hiện ra.
- ✎ Nhấp phải chuột trên thư mục **TP7.0**, chọn **Open** trong menu hiện ra, hoặc nhấp đúp nút trái chuột vào thư mục **TP7.0**. Chúng ta sẽ thấy hình như sau:



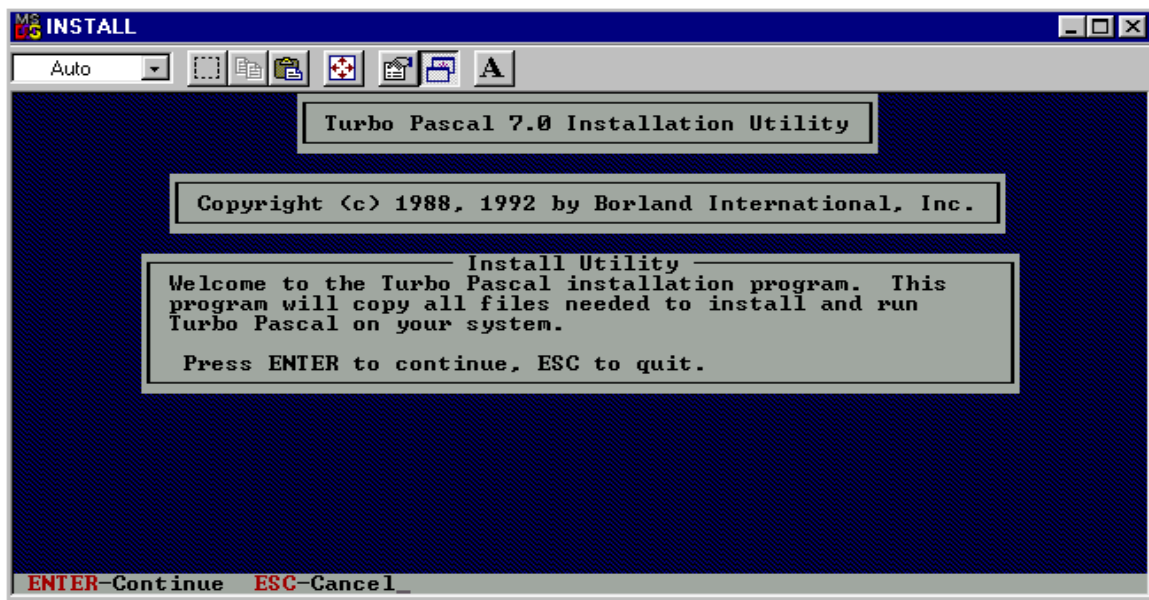
Hình H1 Cửa sổ chứa các thư mục và các tập tin trong thư mục TP7.0

- ✎ Nhấp đúp vào mục **INSTALL**, bạn thấy hộp thoại hiện ra như hình H2.



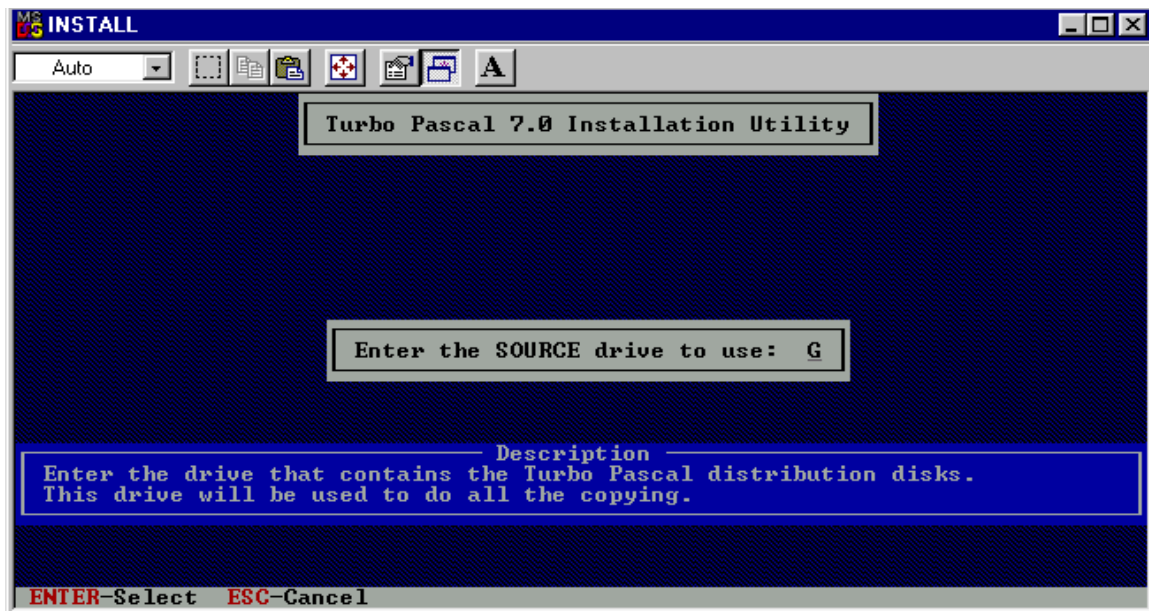
Hình H2

- ✎ Nhấp nút **Run Program**. Bạn sẽ thấy hộp thoại thông báo, trong hộp thoại này, bạn nhấn phím **Enter** để tiếp tục cài đặt, nếu không thích, bạn nhấn phím **ESC** để thoát.



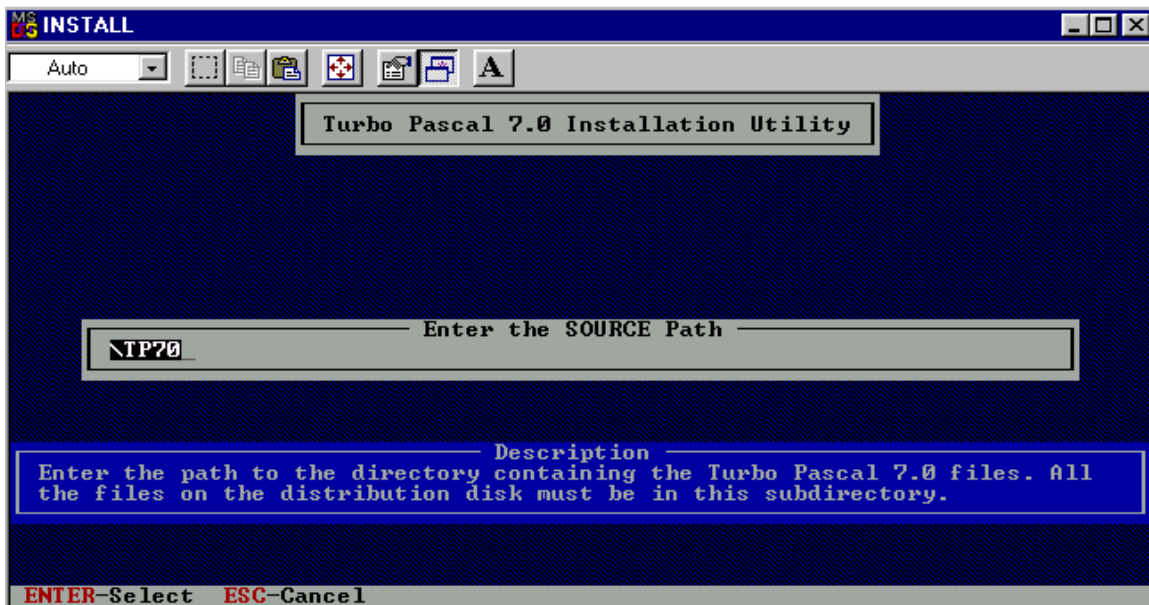
Hình H3 Cửa sổ thông báo cài đặt Turbo Pascal

- ✎ Nhấn phím **Enter** để tiếp tục cài đặt, sẽ có hộp thoại hiện ra với tên ổ đĩa mặc nhiên là tên ổ đĩa CD của máy đang chứa đĩa cài đặt, bạn giữ tên ổ đĩa mặc nhiên này, máy của bạn có thể là tên khác, chúng ta sẽ có hình như sau:



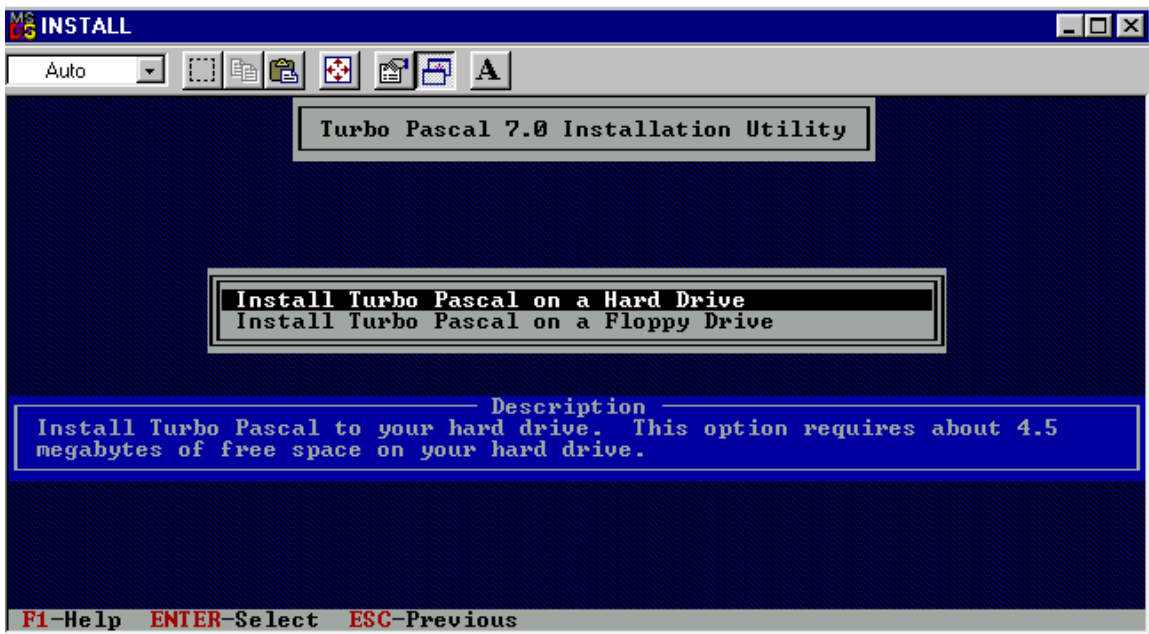
Hình H4 nhập vào tên ổ đĩa chứa Turbo Pascal

✎ Nhấn phím **Enter**, các em sẽ thấy hộp thoại hiện ra như hình H5.



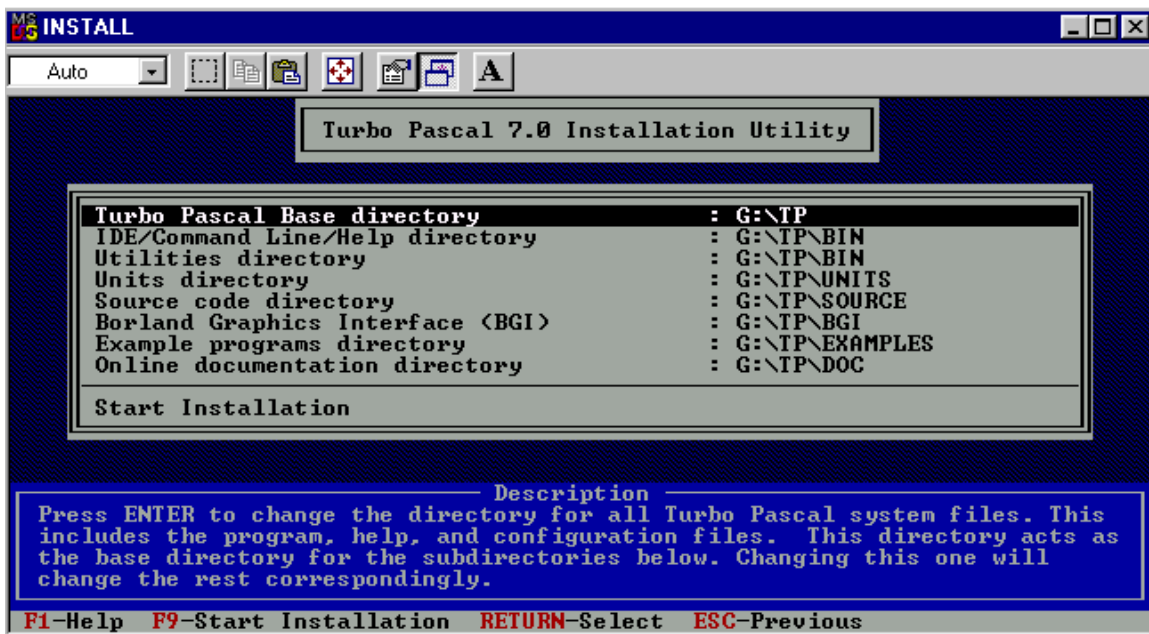
Hình H5 Nhập vào đường dẫn chứa các tập tin của Turbo Pascal 7.0

- ✎ Bạn giữ nguyên đường dẫn mặc nhiên này, máy bạn có thể khác, tùy theo đường dẫn của ổ đĩa CD chứa chương trình.
- ✎ Nhấn phím **Enter**, bạn sẽ thấy hộp thoại hiện ra như hình H6



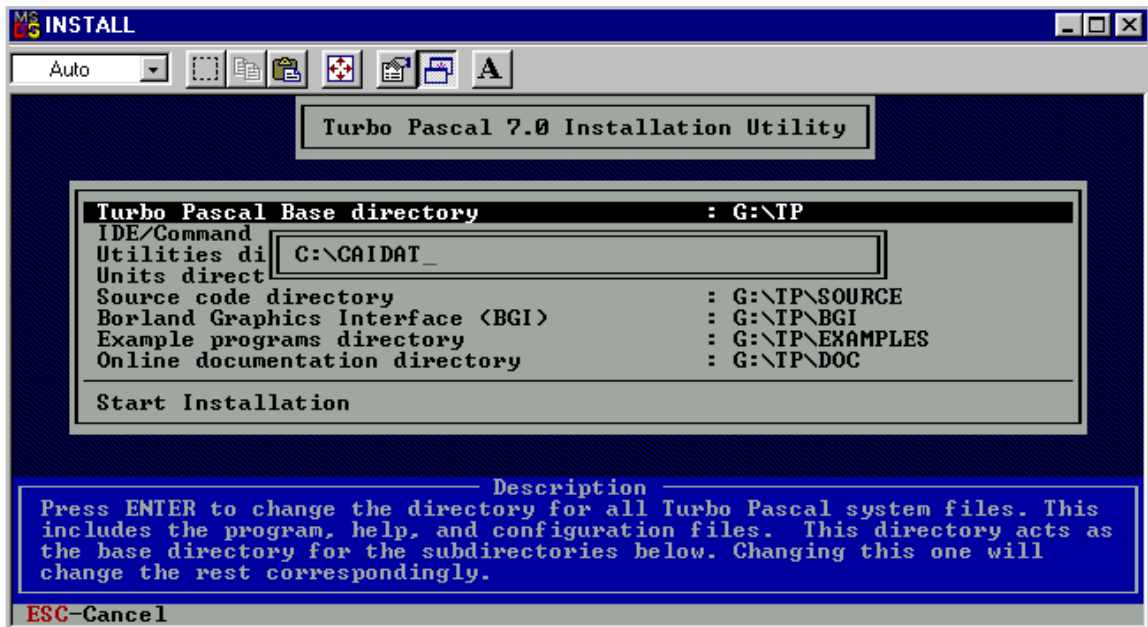
Hình H6 Chọn Install lên ổ đĩa cứng hay đĩa mềm

- ✎ Bạn chọn đĩa cứng (hard drive).
- ✎ Nhấn phím **Enter**, bạn sẽ có



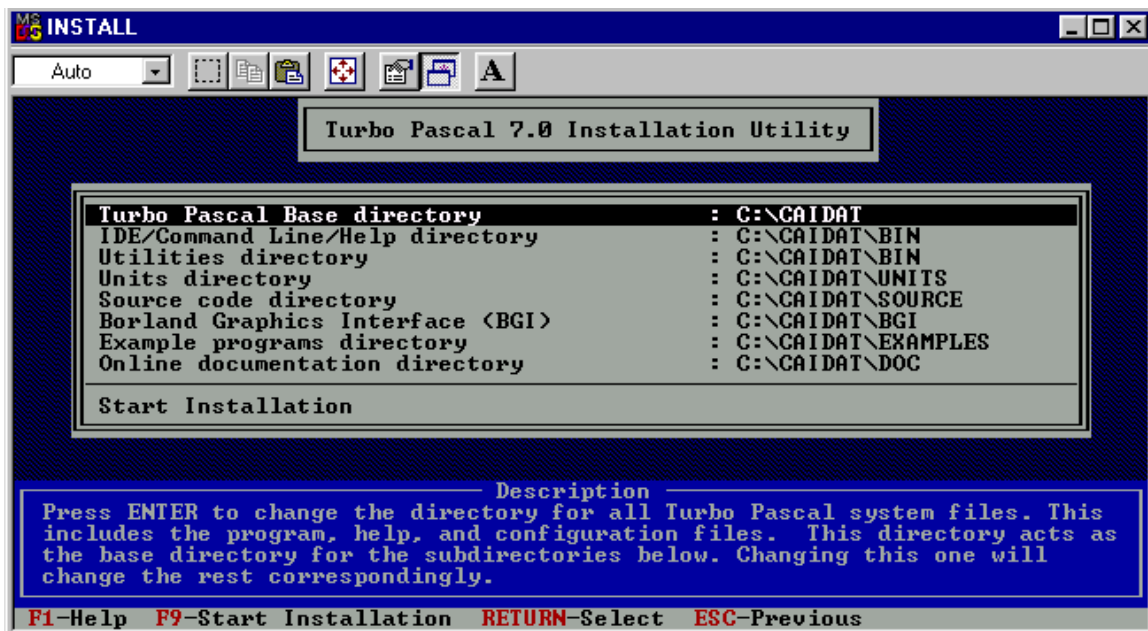
Hình H7 Hình cho chọn lại thư mục hoặc bắt đầu Install

- ✎ Nếu thích thay đổi thư mục chứa tất cả các tập tin hệ thống của Turbo Pascal, bạn nhấn phím **Enter**, nếu không muốn, nhấn phím **F9** để bắt đầu **Install**.
- ✎ Ở đây chúng tôi nhấn phím **Enter**, trong hộp thoại hiện ra, chúng tôi xoá đường dẫn có sẵn và nhập vào **C:\CAIDAT** (nhớ tạo thư mục CAIDAT trước).



Hình H8 Đã sửa lại C:\CAIDAT

- ✎ Nhấn phím **Enter**, bạn sẽ thấy thông tin ổ đĩa và đường dẫn đổi lại như sau:



Hình H9

- ✎ Nhấn phím **F9** để bắt đầu **Install**. Sau một vài thao tác, bạn đã hoàn thành việc cài đặt **Turbo Pascal 7.0** vào ổ đĩa cứng của mình.

2/ Sử dụng chương trình Pascal trên đĩa cứng

a/ Sử dụng đĩa cứng:

- ⇒ Nhấp phải chuột tại menu **Start**, chọn **Explore**.
- ⇒ Trong ổ đĩa **C:** nhấp chọn tại tên **CAIDAT**.
- ⇒ Nhấp đúp chuột tại thư mục **BIN**.
- ⇒ Nhấp đúp chuột tại **Turbo** (hình cửa sổ, đây là Turbo.exe)



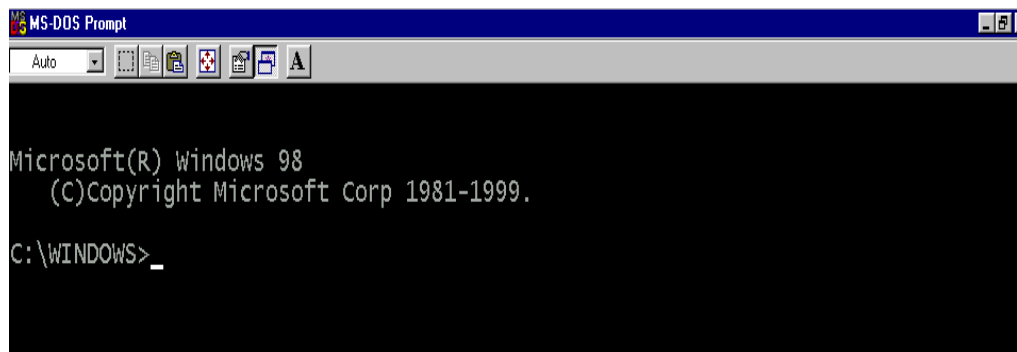
Hình H10

b/ Sử dụng đĩa mềm:

⇒ Với đĩa mềm, các bạn phải có tối thiểu hai tập tin **TURBO.EXE** và **TURBO.TPL**, ở đây các bạn có thể chép 5 tập tin có tên **TUBBO** như ở hình H10. Khi chạy chương trình, bạn chỉ cần nhấp đúp tại Turbo (hình cửa sổ) trong thư mục gốc của ổ đĩa a:\.

c/ Sử dụng MS-DOS PROMPT trong Win9x:

⇒ Vào cửa sổ **MS-DOS PROMPT**, bằng cách nhấp phải nút chuột trên biểu tượng, chọn **Open** trong menu hiện ra, hoặc vào menu **Start**, chọn **Programs**, chọn **MS-DOS PROMPT**. Bạn sẽ thấy hình như sau:



Hình H11 cửa sổ MS-DOS Prompt

⇒ Thông thường sẽ ở thư mục **C:\WINDOWS>** như hình trên.

C:\Windows>**CD** gõ **Enter**. (chuyển về thư mục gốc).

C:\CD **CAIDAT** gõ **Enter**. (Chuyển đến thư mục CAIDAT)

C:\CAIDAT\CD **BIN** gõ **Enter** (chuyển đến thư mục BIN)

C:\CAIDAT\BIN>**Turbo** gõ **Enter**. Cửa sổ Pascal sẽ hiện ra.

Ghi chú: Vì tập tin Turbo.exe nằm tại C:\CAIDAT\BIN\Turbo.exe

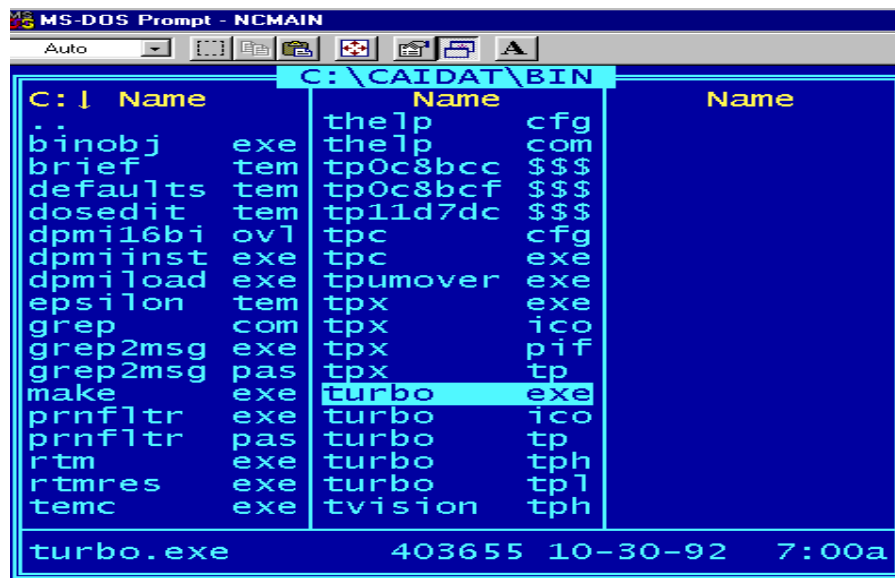
⇒ Hoặc từ thư mục **C:\WINDOWS>** như hình trên.

C:\Windows>**CD** gõ **Enter**. (chuyển về thư mục gốc).

C:\NC gõ Enter hoặc C:\NC\NC gõ Enter. Tùy theo máy của bạn và cách khai báo đường dẫn trong tập tin **Autoexec.bat**.

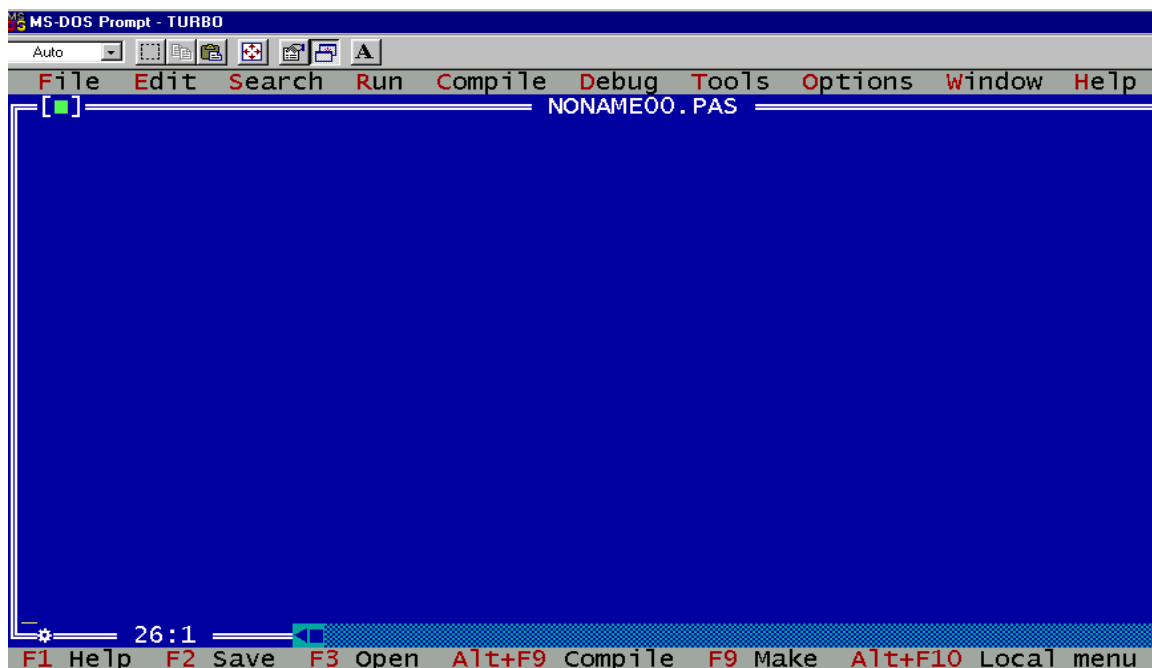
⇒ Vào thư mục chứa tập tin **Turbo.exe**, đưa vệt sáng đến tên tập tin này, gõ phím **Enter**.

Ví dụ: Theo hướng dẫn ở trên, các bạn thấy tập tin **Turbo.exe** nằm trong **C:\CAIDAT\BIN**. Vậy bạn vào thư mục **CAIDAT**, sau đó vào thư mục **BIN** (thực hiện trong **NC**). Bạn sẽ thấy như hình sau:



Hình H12 vào tập tin Turbo.exe

⇒ Gõ phím **Enter**, cửa sổ PASCAL hiện ra cho bạn lập trình.



Hình H13 cửa sổ Pascal

III/ Cách sử dụng Turbo Pascal

1/ Tạo - lưu - mở tập tin chương trình

a/ Tạo tập tin

⇒ Bạn vào cửa sổ Pascal, giả sử nhập đoạn chương trình sau:

Program Hienthi;

Begin

 Write ('chung toi se giai thich chuong trinh nay trong cac bai sau');

 Readln;

End.


```

MS-DOS Prompt - TURBO
Auto
File Edit Search Run Compile Debug Tools Options Window Help
[ ] NONAME00.PAS
Program Hienthi;
Begin
  Write('Chung toi se giai thich chuong trinh nay trong cac bai sau');
  Readln;
End.

```

Hình H14 Cửa sổ khi viết chương trình xong

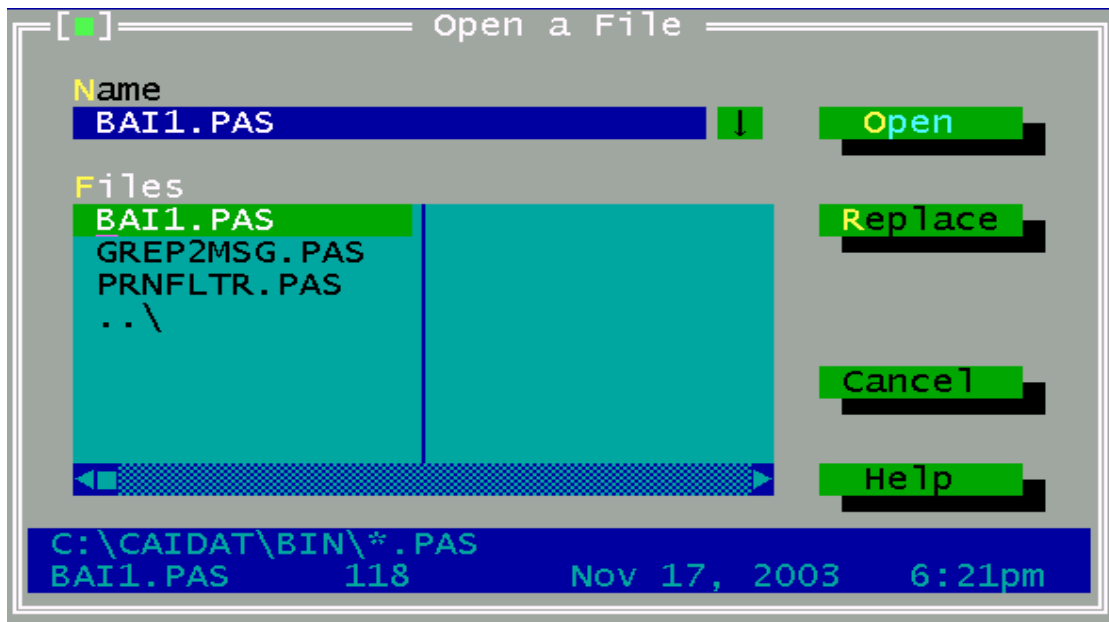
b/ Lưu tập tin chương trình trên

- ⇒ Nhấn phím **F2** hoặc vào menu **File**, chọn **Save**.
- ⇒ Chọn đường dẫn và tên tập tin cần lưu. Thường Pascal sẽ lấy đường dẫn chứa tập tin Turbo.exe, vậy trong bài này sẽ là C:\CAIDAT\BIN. Bạn nhập tên Bai1. Pascal sẽ tự động gán phần mở rộng là .PAS. Nếu muốn lưu nơi khác, bạn nhập đường dẫn và tên tập tin. Nếu sau này muốn lưu với một tên khác, tập tin cũ vẫn còn, bạn vào menu File, chọn Save as.

Ví dụ: E:\Bai1, D:\Luu\Bai1...

c/ Mở tập tin chương trình

- ⇒ Vào cửa sổ **Pascal**.
- ⇒ Vào menu **File**, chọn **Open** hoặc nhấn phím **F3**, trong hộp thoại hiện ra, chọn **Bai1**, nhấp nút **Open**. Bạn sẽ thấy chương trình lúc này hiện ra như hình H14 ở trên.



Hình H15 Cửa sổ mở chương trình

2/ Sử dụng các phím trong soạn thảo Program

a/ Dịch chuyển con trỏ

- ⇒ Các phím hướng **↑**, **↓**, **→**, **←** trên bàn phím dịch chuyển dấu nháy lên dòng trên, xuống dòng dưới, qua phải 1 kí tự, qua trái 1 kí tự.
- ⇒ **Ctrl + →** và **Ctrl + ←**: sẽ dịch chuyển dấu nháy theo từng từ.
- ⇒ **Home**: Đưa dấu nháy về đầu dòng.
- ⇒ **End**: Đưa dấu nháy về cuối dòng.
- ⇒ **Page Up** hoặc **Page Down**: Dịch dấu nháy lên hoặc xuống theo từng trang màn hình.

⇒ **Ctrl + Page Up** hoặc **Ctrl + Page Down**: Đưa dấu nháy về đầu tập tin hay cuối tập tin.

b/ *Sửa chữa văn bản*

⇒ Phím **Del** để xóa một kí tự tại vị trí hiện thời của dấu nháy.

⇒ Phím **Backspace** để xóa kí tự nằm bên trái dấu nháy.

⇒ **Ctrl + Y** để xóa cả dòng đang chứa dấu nháy.

⇒ **Insert** để chọn chế độ chèn hoặc đè.

c/ *Các phím chức năng*

Nhìn vào cửa sổ soạn thảo chương trình của Pascal, bạn thấy các phím chức năng sẽ có tác dụng như sau:

⇒ **F1** Help: Nhấn phím F1 sẽ hiện thông tin chỉ dẫn.

⇒ **F2** Save: Nhấn phím F2 sẽ lưu chương trình.

⇒ **F3** Open: Nhấn phím F3 sẽ mở chương trình.

⇒ **Alt+F9** Compile: Nhấn giữ phím Alt, nhấn thêm phím F9 sẽ Compile chương trình.

⇒ **F9** Make: Compile chương trình.

⇒ **Alt+F10** Local menu: Hiện ra menu.

⇒ **Alt+Kí tự có màu đỏ** đứng trước các mục menu: Xuất hiện menu đó

⇒ **ESC**: Nhấn phím ESC sẽ làm biến mất các hộp thoại, menu... đang hiện trên màn hình soạn thảo.

3/ Biên dịch (Compile) chương trình

Khi viết xong một chương trình, để biên dịch, bạn có thể thực hiện một trong các cách sau:

⇒ Nhấn **Alt+F9**

⇒ Nhấn **Alt+C**, dùng phím **↓**, **↑** di chuyển vệt sáng đến chữ **Compile** trong menu, nhấn phím **Enter**. Hoặc nhấp chuột tại menu **Compile**, nhấp chọn mục **Compile** trong menu buông xuống, hoặc nhấn phím **F10**, dùng phím hướng di chuyển vệt sáng đến mục **Compile** trên menu, chọn **Compile** trong menu buông xuống.

❑ Ở chế độ mặc nhiên, Pascal sẽ chọn **Destination là Memory**. Trong trường hợp này, khi bạn nhấn Alt+F9 hoặc vào menu chọn Compile thì sẽ biên dịch chương trình, báo kết quả đúng sai. Không tạo tập tin gì cả.

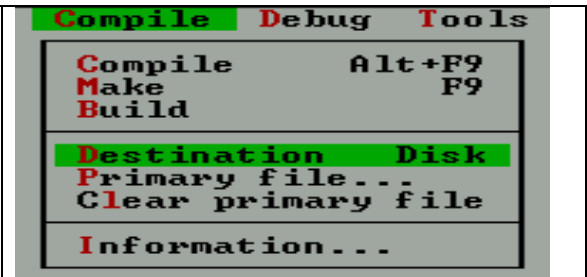
Khi nhấp chuột tại menu Compile.
Bạn thấy **Destination là memory**.



Hình H16

❑ Nếu muốn chọn chế độ **Disk**, bạn chỉ cần nhấp chuột tại dòng **Destination Memory** hoặc nhấn phím **D** trên bàn phím, menu biến mất. Thực hiện chọn lại menu **Compile**, bạn sẽ thấy như hình sau:

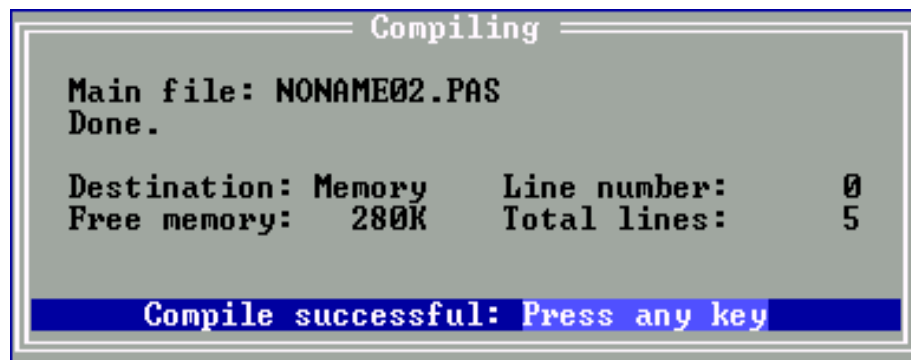
Khi nhấp chuột tại menu Compile.
Bạn thấy **Destination** là **Disk**.



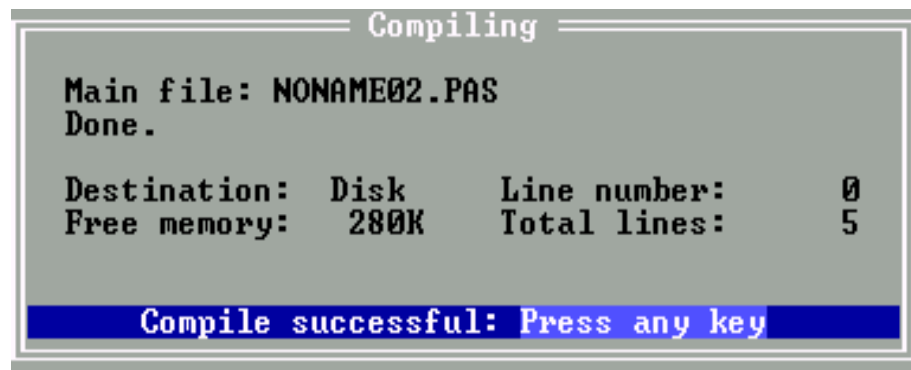
Hình H17

Bạn lưu ý, nếu **Compile** chương trình khi **Destination** là **Disk** thì sẽ tạo tập tin với tên hiện tại có phần mở rộng là **.EXE** trên đĩa, bạn có thể chạy chương trình này bằng cách gõ đường dẫn và tên chương trình ngay đầu nhắc của hệ thống, có thể chạy bằng cách chọn tên chương trình trong Norton Commander, có thể chạy trong menu Start/Run của Windows.

- Nếu không có chỗ sai, bạn sẽ thấy hộp thông báo như sau



Hình H18 Compile khi chọn Destination là Memory



Hình H19 Compile khi chọn Destination là Disk

- Nếu sai, bạn sẽ thấy một dòng chữ màu đỏ báo phía đỉnh trên màn hình.

4/ Thoát khỏi Pascal

- ⇒ Vào menu **File**, chọn **Exit** hoặc
- ⇒ Nhấn **Alt+X**

Ghi chú:

Chạy thử chương trình, các bạn sẽ biết ở bài sau



TÓM TẮT

- ✎ Để cài đặt Turbo Pascal 7.0, bạn phải có đĩa CD hoặc bộ đĩa mềm chứa Turbo 7.0. Biết thực hiện các thao tác để tìm đến tập tin có tên Install.exe, thực hiện các tác động trên hộp thoại hiện ra.
- ✎ Để thực hiện chương trình Pascal trên đĩa cứng, bạn chỉ cần tìm đến thư mục có chứa tập tin Turbo.exe, thông thường là thư mục BIN. Nhấn đúp vào tên tập tin này hoặc nhấp phải chuột tại tên tập tin này, chọn Open trong menu hiện ra. Để thực hiện trên đĩa mềm cũng tương tự như vậy, nếu dùng đĩa mềm, ta chỉ cần chép 5 tập tin có tên TURBO. Để thực hiện trên MS-DOS PROMPT, bạn vào MS-DOS PROMPT, trên màn hình DOS, vào Norton Commander, chọn thư mục có chứa tập tin Turbo.exe, di chuyển vệt sáng đến tập tin này và nhấn phím Enter.
- ✎ Để tạo một chương trình, bạn gõ thông tin vào cửa sổ soạn thảo.
- ✎ Để lưu chương trình, bạn nhấn phím F2 hoặc vào menu File, chọn Save . Sau này muốn lưu thêm một tập tin nữa có tên khác, bạn vào menu File, chọn Save as.
- ✎ Để mở một chương trình, bạn nhấn phím F3 hoặc vào menu File, chọn Open, trong hộp thoại hiện ra, chọn tên tập tin muốn mở, nhấp nút Open trên hộp thoại.
- ✎ Cần thao tác các phím chức năng như hướng dẫn ở bài học để thực hiện cho công việc soạn thảo chương trình.
- ✎ Biết các cách biên dịch chương trình.



BÀI TẬP THỰC HÀNH

- ❶ Tạo một thư mục có tên PASCAL trong thư mục gốc của ổ đĩa C:\. Cài đặt Turbo Pascal 7.0 vào trong thư mục này.
- ❷ Xác định tập tin Turbo.exe nằm ở đâu khi đã cài đặt như trên.
A/ Nằm trong C:\
B/ Nằm trong C:\PASCAL
C/ Nằm trong C:\PASCAL\BIN
D/ Cả 3 câu trên đều sai.
- ❸ Sau khi đã cài đặt xong, chép 5 tập tin có tên TURBO trong thư mục BIN vào thư mục gốc của ổ đĩa mềm A:\
- ❹ Thực tập vào PASCAL với 3 cách:
A/ Thực hiện trên ổ đĩa cứng.
B/ Thực hiện trên ổ đĩa mềm.
C/ Thực hiện bằng cách sử dụng MS-DOS PROMPT.
Hãy trình bày ra giấy các bước để thực hiện được theo 3 cách trên. Theo bạn cách nào là tiện lợi và nhanh nhất.
- ❺ Tạo một chương trình như sau:
Program Thuchanh;
Begin
 Write('Ban co thich hoc PASCAL không?');
 Readln;
End.
A/ Thử biên dịch chương trình bằng các cách đã nêu trong bài học.
B/ Xoá dấu chấm phẩy (;) phía sau chữ Thuchanh. Biên dịch chương trình để xem kết quả. Hãy thử bỏ dấu chấm sau End để xem thông báo khi biên dịch, thử bỏ dấu nháy đơn bao chuỗi 'Ban co thich hoc PASCAL không'. Thử bỏ Begin và End. Sau khi thử xong, sửa lại cho đúng.
C/ Hãy lưu chương trình này trong thư mục BTPASCAL của ổ đĩa C:
D/ Thoát khỏi PASCAL.
- ❻ Tập mở chương trình này bằng các cách đã học và lưu với một tên khác, cũng chọn địa chỉ lưu là C:\BTPASCAL.
Khi lưu tập tin với tên khác như trên, chúng ta sẽ thấy:
A/ Tập tin cũ sẽ được thay đổi thành tên mới.
B/ Pascal sẽ báo sai.
C/ Tập tin cũ vẫn còn, có thêm tập tin mới với tên vừa lưu.
D/ Cả 3 câu trên đều sai.
- ❼ Tạo một chương trình như bài 5, Biên dịch với Destination là Disk. Sau khi có tên chương trình với phần mở rộng là .EXE (sẽ nằm trong C:\CAIDAT\BIN). Chạy thử chương trình trên bằng 3 cách: Chạy bằng cách gõ đường dẫn và tên chương trình tại dấu nhắc của hệ điều hành. Vào NC để thực hiện. Thực hiện tại menu Start/Run của Windows.

CẤU TRÚC CHƯƠNG TRÌNH PASCAL

KIẾN THỨC YÊU CẦU

- Biết được các ký tự sử dụng trong Pascal.
- Biết cấu trúc của một chương trình Pascal.
- Biết phân tích một vấn đề, chọn giải thuật để giải quyết vấn đề.
- Biết cách dịch chương trình sang mã máy.
- Biết chạy thử chương trình và kiểm tra kết quả.

I/ Những nguyên tắc cơ bản của lập trình Pascal

1/ Các ký hiệu sử dụng trong Pascal

Bất cứ một ngôn ngữ lập trình nào cũng được xây dựng từ một số hữu hạn các ký hiệu. Trong ngôn ngữ Pascal được xây dựng từ các ký hiệu:

Begin, end, var, while, do, {, }, ;, ...

Và các kí tự **a, b, c, d, ..., A, B, C, D, ..., 1, 2, 3, 4, ...**

Ngôn ngữ Pascal không dùng các ký hiệu của bộ chữ hy Lạp.

Để xây dựng thành chương trình, các ký hiệu phải tuân theo những quy ước về ngữ pháp và ngữ nghĩa quy định của Pascal.

a/ Danh hiệu (identifiler)

Trong Pascal, để đặt tên cho các **biến, hằng, kiểu, chương trình con** ... Ta dùng **danh hiệu** (identifiler). Danh hiệu trong Pascal quy định bắt đầu phải là một chữ cái, sau đó có thể là chữ cái, chữ số sau là dấu gạch dưới “_”.

Ví dụ:

Tam
X
PT_bac_1
Delta
Z200

Ví dụ: các biến sau không phải là danh hiệu

2bien
n!
Bien x

Trong Pascal danh hiệu không phân biệt chữ thường và chữ hoa.

Ví dụ: y với Y là một. Thanh_Da và THANH_dA là một.

Chú ý: Chúng ta không nên đặt danh hiệu trùng với danh hiệu của ngôn ngữ và nên dùng danh hiệu có tính gợi nhớ để dễ viết và kiểm tra chương trình, người đọc cũng cảm thấy dễ hiểu.

Ví dụ: Write, Writeln, read, readln, sqrt, integer, real ...

b/ Từ khoá (key word)

Trong ngôn ngữ có những từ được dành riêng như là những phần tử tạo nên ngôn ngữ. Do đó chúng ta không được đặt những danh hiệu trùng với những từ dành riêng này. người ta thường gọi những từ này là từ khoá (key word).

Ví dụ: **Program, begin, end, while, do, procedure, function, type, var ...**

Từ dành riêng này cũng không phân biệt chữ thường và chữ hoa.

Một số từ dành riêng trong Pascal

And	Array	Begin	Case	Const	Div	Do	Downto
Else	End	File	For	Forward	Function	Goto	If
In	Label	Mod	Nil	Not	Of	Or	Procedure
Program	Record	Repeat	Set	Shl	Shr	String	Then
To	Type	Until	Var	While	With	Xor	

c/ Khoảng trắng, dấu chấm phẩy, dấu móc, dấu nháy đơn, toán tử, toán hạng.

⇒ Khoảng trắng “ ”: Dùng để cách biệt 2 từ trong chuỗi.

⇒ Dấu chấm phẩy “;”: Dùng sau tên chương trình, sau các câu lệnh. (trừ câu lệnh trước Else, chúng tôi sẽ đề cập sau)

⇒ Dấu móc “{ }”: Trong Pascal những gì đặt trong hai dấu móc sẽ là phần ghi chú.

⇒ Dấu nháy đơn “'” : Dùng để bao một chuỗi.

⇒ Toán tử: Đặt giữa hai toán hạng. Ví dụ: $x+2$, $y=8$, $7>3$, $x<7$, $z>=y$...
vậy +, =, >, <, <= là các toán tử, còn hai bên sẽ là các toán hạng.

2/ Cấu trúc một chương trình Pascal

Một chương trình trong Pascal gồm các phần khai báo và sau đó là thân của chương trình.

⇒ Khai báo **Program**

⇒ Khai báo **Uses**

⇒ Khai báo **Label**

⇒ Khai báo **Const**

⇒ Khai báo **Type**

⇒ Khai báo **Var**

⇒ Khai báo các chương trình con (thủ tục hay hàm)

⇒ Thân chương trình

Thân của chương trình được bắt đầu bằng từ khoá **Begin** và kết thúc bằng từ khoá **End** và dấu chấm “.”. Giữa **Begin** và **End.** là các phát biểu.

Ví dụ:

Program Chuongtrinhmau;

Uses

.....

Label

.....

Const

.....

Type

.....

Var

..... (Khai báo tên và kiểu của các biến)

Function ...

End;

Procedure ...

End;

Begin

.....

.....

End.

Thông thường trong một chương trình Pascal, các khai báo Uses, Label, const, type, Function, Procedure có thể có hoặc không tùy theo bài, nếu không dùng biến thì cũng không cần khai báo Var (như ví dụ ở bài 1), tuy nhiên hầu hết các chương trình đều dùng khai báo Program, var các biến và thân chương trình.

II/ các bước cơ bản để viết một chương trình máy tính

1/ Phân tích công việc và chọn cách giải, gọi là giải thuật (algorithm)

Để lập trình giải quyết một vấn đề, điều trước tiên cần có là giải thuật. *Giải thuật là kiến thức về lĩnh vực của vấn đề đang giải quyết.* Giải thuật có ý nghĩa quyết định đến thành công của chương trình, có giải thuật tốt, mới có chương trình tốt. Có những giải thuật tổng quát như tổ chức cấu trúc dữ liệu, thuật toán sắp xếp, thuật toán tìm kiếm, thuật toán phân phối bộ nhớ ... những thuật toán thuộc lĩnh vực chuyên môn như những giải thuật của toán, của cơ sở dữ liệu, của trí tuệ nhân tạo, của đồ họa ... Tuy nhiên trong lĩnh vực cấp 2, các bạn chỉ cần biết qua các thuật toán đơn giản.

Ví dụ: Để giải phương trình bậc 2, ta phải xét dấu delta với ($\Delta = b^2 - 4 \cdot a \cdot c$)

2/ Viết chương trình bằng ngôn ngữ Pascal

⇒ Quan trọng nhất là phần cốt lõi của thân chương trình để giải quyết thành công yêu cầu đề ra.

⇒ Sau đó là phần nhập dữ liệu ở đầu chương trình và xuất kết quả ở cuối chương trình.

⇒ Cuối cùng là thêm phần khai báo, cần dùng những biến nào khai báo trong phần Var, đặt tên chương trình trong phần khai báo program.

Ví dụ: Để giải phương trình bậc nhất là phát biểu If với điều kiện là các trường hợp a bằng hay khác 0, b bằng hay khác 0. Trong phần lõi thường không có nhập xuất.

Tóm lại: Khi viết một chương trình, đầu tiên đừng nghĩ tên chương trình là gì, dùng các biến nào, khoan nghĩ đến phải nhập xuất dữ liệu như thế nào cho đẹp mắt, mà phải tập trung trước tiên vào việc viết phần lõi của chương trình sao cho thể hiện chính xác qua giải thuật.

3/ Dịch chương trình sang mã máy

Biên dịch chương trình xem có chỗ nào viết sai về cú pháp hay một vấn đề nào đó để điều chỉnh lại.

4/ Chạy thử và kiểm tra kết quả

Khi biên dịch không báo lỗi, chưa chắc chương trình của bạn cho kết quả đúng, nhiều khi giải thuật sai, sẽ cho kết quả sai. Do đó bạn chạy thử chương trình, kiểm tra kết quả xem có chính xác hay không.

Ví dụ: chỗ đó thay vì dấu cộng bạn lại cho dấu trừ, chỗ đó là biến a, bạn lại để là biến b. Cú pháp đúng, biên dịch đúng nhưng kết quả sai.

⇒ Để chạy chương trình, bạn thực hiện một trong những cách sau:

■ Nhấn **Ctrl+F9** hoặc

■ Nhấn **Alt+R** (hoặc nhấp chuột trái tại tên menu, hoặc nhấn phím F10, di chuyển vệt sáng đến mục Run), đưa vệt sáng đến **Run**, nhấn phím **Enter**.



TÓM TẮT

👉 Cần nắm vững những nguyên tắc cơ bản của lập trình Pascal, biết các ký hiệu dùng được, không dùng được, không nên dùng. Biết một số các từ khoá để không dùng tên danh hiệu giống như vậy.

- ✎ Cần biết các kí tự thường dùng như dấu chấm phẩy, dấu nháy đơn, dấu móc, và chúng thường dùng ở đâu.
- ✎ Cần biết các toán tử trong Pascal.
- ✎ Biết cấu trúc của một chương trình Pascal, vị trí của chúng, phần nào có thể bỏ được, phần nào không thể thiếu.
- ✎ Trước khi khai báo các biến, phải có từ khoá Var.
- ✎ Biết bước đầu tiên khi viết một chương trình, trong chương trình, biết phần nào là cốt lõi để tập trung thực hiện trước.
- ✎ Biết biên dịch và sửa lỗi chương trình.
- ✎ Qua kết quả, kiểm tra lại giải thuật, sử dụng biến đúng hay không...



BÀI TẬP THỰC HÀNH

- ❶ Xác định các khai báo biến như sau là đúng hay sai.
 - A/ ThanhTien
 - B/ 1_Luong
 - C/ Dem so
 - D/ !Giaithua
 - E/ \$USD
 - F/ Ket – Qua
 - G/ Ket_qua
 - H/ BaSo555
 - I/ Nam 2003
- ❷ Danh hiệu nào sau đây dùng không được
 - A/ begin
 - B/ Batdau
 - C/ Until
 - D/ DenKhi
- ❸ Trong Pascal, nếu dùng dấu nháy bao chuỗi cho câu sau thì có đúng không.
“Toi rat thich hoc ngon ngu PASCAL”
- ❹ Tìm chỗ sai và thiếu trong các chương trình sau:

```
Program Quangcao  
  Write('Chao mung SEA Games 22 tai Viet Nam').  
  Readln;  
End
```

```
Program Vui:  
X : integer; { Cho biết khai báo đúng}  
Y : Real; (Cho biết khai báo đúng)  
Begin  
  Write("Khai bao bien x co kieu nguyen, bien Y co kieu so thuc");  
  Readln.  
End;
```

```
Program Nhanxet;  
{Begin}  
  Write(Hoc Pascal kho qua!);  
  Readln;  
End.
```

- 5 Tìm giải thuật tính tổng 10 số nguyên đầu tiên.
- 6 Tìm giải thuật tính tổng 5 số chẵn đầu tiên.
- 7 Tìm giải thuật tính Max của 2 số, max của 3 số.
- 8 Tìm giải thuật để biết ước số chung lớn nhất của 2 số.
- 9 Tìm giải thuật để biết một số có phải là số nguyên tố hay không.
- 10 Tìm giải thuật để giải phương trình bậc nhất.

KIỂU DỮ LIỆU CƠ SỞ

KIẾN THỨC YẾU CẦU

- Nhận biết một số thể hiện của màn hình Windows.
- Biết cách sử dụng các thành phần của màn hình Windows.
- Luyện tập thao tác sử dụng chuột.

I/ Dữ liệu kiểu số

1/ Khái niệm về dữ liệu và kiểu dữ liệu

Trong Pascal một dữ liệu (data) bao giờ cũng thuộc về một kiểu dữ liệu nhất định. Một kiểu dữ liệu (data type) là một sự quy định về hình dạng, cấu trúc và giá trị của dữ liệu cũng như cách biểu diễn và cách xử lý dữ liệu. Ngôn ngữ lập trình chỉ chấp nhận xử lý những dữ liệu tuân theo sự quy định về kiểu của ngôn ngữ lập trình đó.

2/ Các kiểu dữ liệu trong Pascal

Có thể phân loại như sau:

a/ Các kiểu đơn giản (simple type):

- ⇒ Kiểu real
- ⇒ Các kiểu rời rạc: integer, char, boolean, byte, word, liệt kê, miền con.

b/ Các kiểu có cấu trúc:

- ⇒ Array (dãy)
- ⇒ Record (mẫu tin)
- ⇒ Set (tập hợp)
- ⇒ File (tập tin)
- ⇒ String (chuỗi)

c/ Kiểu con trỏ (pointer type)

3/ Tìm hiểu một số kiểu căn bản

a/ Kiểu số nguyên (integer): Là các số nguyên, biểu diễn dưới dạng số hệ thập phân (cơ số 10 dùng các số từ 0 đến 9), nó cũng có thể biểu diễn dưới

dạng hệ thập lục phân hexadecimal (cơ số 16, dùng các số từ 0 đến 9, và A, B, C, D, E, và F), dùng dấu \$ đặt trước số dạng cơ số 16, nằm trong khoảng từ \$0 đến \$FFFF. Có 5 kiểu số nguyên như sau:

Ví dụ:

Hexidecimal	Decimal
\$FFFF	65535
\$1A3E	6718
\$321F	12831

Tên kiểu	Phạm vi	Kích thước
Shortint	-128 đến 127	1 byte
Integer	-32768 đến 32767	2 byte
Longint	-2147483648 đến 2147483647	4 byte
Byte	0 đến 255	1 byte
Word	0 đến 65535	2 byte

⇒ Cách khai báo: Tên biến, dấu hai chấm “:”, kiểu, dấu chấm phẩy “;”

Ví dụ:

VAR

X : Byte;

So : Integer;

Ketqua : Integer;

Chúng ta có thể khai báo nhiều biến cùng kiểu cách nhau bằng dấu phẩy “,”

Ví dụ:

VAR

So, Ketqua : Integer;

⇒ Các phép toán có thể thực hiện trên kiểu số nguyên: + (cộng), - (trừ), * (nhân), DIV (phép chia nguyên) và MOD (số dư).

b/ Kiểu số thực (real): Là những dữ liệu số thực, được viết dưới dạng một số thập phân, có thể theo sau bằng một lũy thừa của 10 (ký hiệu bằng chữ E)

Ví dụ:

-234.5678

-2.5E2 có nghĩa là -2.5×10^2

1.8E-10 có nghĩa là 1.8×10^{-10}

Tên kiểu	Phạm vi	Kích thước
Single	1.5E-45 đến 3.4E+38	4 byte
Real	2.9×10^{-39} đến 1.7×10^{38}	6 byte
Double	5.0E-324 đến 1.7E+308	8 byte
Extended	3.4E-4932 đến 1.1E+4932	10 byte
Comp	-9.2E+18 đến 9.2E+18	8 byte

⇒ Cách khai báo: Tên biến, dấu hai chấm “:”, kiểu, dấu chấm phẩy “;”

Ví dụ:

VAR

X : Real;

Y : Single;

Z : Double;

Chúng ta có thể khai báo nhiều biến cùng kiểu cách nhau bằng dấu phẩy “,”

Ví dụ:

VAR

So, Ketqua : Real;

⇒ Các phép toán có thể thực hiện trên kiểu số nguyên: + (cộng), - (trừ),

* (nhân), / (chia).

III/ Dữ liệu kiểu Logic

1/ Kiểu Logic (Boolean): Gồm hai giá trị False (sai) và True (đúng).

Ví dụ:

VAR

BienLogic : Boolean;

Khi khai báo xong ta có thể gán cho BienLogic một trong hai giá trị:

BienLogic := True; hay BienLogic := False;

2/ Biểu thức Logic

Ví dụ: $x > 5$; $(x < 4) \text{ And } (y > 7)$

▪ $x > 5$ sẽ đúng (True) khi x có giá trị từ 6 trở lên, và sai (False) khi x có giá trị nhỏ hơn hoặc bằng 5.

▪ $(x < 4) \text{ And } (y > 7)$ sẽ đúng khi x có giá trị nhỏ hơn 4 và giá trị của y từ 8 trở lên.

Kết quả của biểu thức trên sẽ có giá trị là đúng hoặc sai. Các biểu thức này thường dùng làm điều kiện trong các phát biểu của Pascal.

Bạn thường dùng 6 toán tử so sánh đó là: = , > , < , >= , <= , <>

Ví dụ:

If $(x > 10)$ Then

If $(x > 2) \text{ And } (y > 2)$ Then

Đối với kiểu này, các bạn cần biết các toán tử luận lý NOT, AND, OR và XOR.

A	NOT A
True	False
False	True

A	B	A AND B
True	True	True
True	False	False
False	True	False
False	False	False

A	B	A OR B
True	True	True
True	False	True
False	True	True
False	False	False

A	B	A XOR B
True	True	False
True	False	True
False	True	True
False	False	False

III/ Dữ liệu kiểu Char (ký tự)

1/ Kiểu Char: Là những dữ liệu ký tự, một ký tự được viết trong hai dấu nháy (' ').

Ví dụ: '3', 'M', 'N', 'a', 'b'

VAR

Kytu : Char;

Kytu := 'A';

Một kí tự được chứa trong một byte.

Kí tự được biểu diễn trong bộ nhớ bởi giá trị của nó trong bảng mã ASCII.

Ví dụ: Ký tự 'B' có mã ASCII là 66, sẽ được biểu diễn trong bộ nhớ bằng 1 byte có trị là 66.

Như vậy các ký tự được biểu diễn bằng các trị từ 0 đến 255.

2/ *Toán tử và các hàm thư viện dùng cho kiểu Char*

Các ký tự có thể so sánh với nhau dựa trên bảng mã ASCII. Vậy có thể dùng các toán tử so sánh đối với kiểu này.

Ví dụ: 'A' < 'B' vì trong bảng mã ASCII A=65 và B=66 vì 65<66.

Hàm **CHR(n:Byte)**: Hàm này cho ta ký tự có mã ASCII bằng n.

Ví dụ: CHR(65) = 'A', CHR(97) = 'a', CHR(67) = 'C', CHR(98) = 'b'

Hàm **ORD(c:Char)**: Hàm này cho ta mã ASCII của ký tự c.

Ví dụ: ORD('A') = 65, ORD('a') = 97

IV/ Dữ liệu kiểu String (chuỗi)

1/ *Kiểu String*: Là một chuỗi các ký tự, chiều dài tối đa là 255.

Ví dụ:

VAR

Chuoi : String;

Chuoi := '***CHAO BAN ***';

Bạn cũng có thể giới hạn cho chuỗi.

Ví dụ:

VAR

Chuoingan : String[8];

Chuoingan := 'ThanhDa';

Nếu khai báo biến như trên, bạn dùng chuỗi có chiều dài tối đa là 8.

2/ *Toán tử nối chuỗi*: Bạn có thể dùng dấu cộng '+' để nối hai chuỗi với nhau

Ví dụ:

VAR

Chuoi : String;

Chuoimot : String;

Chuoihai : String[6];

Bạn có thể thực hiện như sau:

Chuoimot := 'Truong DHBK';

Chuoihai := 'TP.HCM'

Chuoi := Chuoimot + Chuoihai;

VI/ Một số hàm và thủ tục thư viện của Pascal

1/ **Hàm**

□ **Hàm ABS(Num)**

▪ Num có kiểu là Integer hoặc Real.

▪ Hàm này cho kết quả là trị tuyệt đối của Num, kiểu kết quả giống kiểu đối số Num.

Ví dụ: ABS(-34)=34; ABS(-34.5) = 3.4500000000E+01

□ **Hàm EXP(Num)**

▪ Num có kiểu là Integer hoặc Real.

▪ Kết quả là e mũ Num (e^{Num}), kết quả sẽ có kiểu Real.

Ví dụ: EXP(2) = e^2

□ **Hàm Ln(Num)**

▪ Num có kiểu là Integer hoặc Real.

▪ Kết quả là Logarit tự nhiên của Num. Kiểu kết quả là Real.

Ví dụ: $\text{Ln}(10) = 2.3025850930\text{E}+00$

❑ Hàm **SQR(Num)**

- Num có kiểu là Integer hoặc Real.
- Kết quả là bình phương của Num. Kiểu kết quả là kiểu của đối số.

Ví dụ: $\text{SQR}(3) = 9$, $\text{SQR}(1.2) = 1.4400000000\text{E}+00$

❑ Hàm **SQRT(Num)**

- Num có kiểu là Integer hoặc Real.
- Kết quả là căn bậc hai của Num (Num phải lớn hơn 0), kết quả có kiểu là Real.

Ví dụ: $\text{SQRT}(4) = 2.0000000000\text{E}+00$

❑ Hàm **INT(Num)**

- Num có kiểu là Integer hoặc Real.
- Kết quả cho ta phần nguyên của Num. Kiểu kết quả là Real.

Ví dụ: $\text{INT}(4.6) = 4.0000000000\text{E}+00$

❑ Hàm **FRAC(Num)**

- Num có kiểu là Integer hoặc Real.
- Cho ta phần lẻ của đối số. Kiểu kết quả là Real.

Ví dụ: $\text{FRAC}(4.3) = 0.3$, $\text{FRAC}(-2.5) = -0.5$

❑ Hàm **ROUND(Num)**

- Num có kiểu là Real.
- Kết quả làm tròn số theo nguyên tắc thông thường, kiểu kết quả là Integer.

Ví dụ: $\text{ROUND}(3.6) = 4$, $\text{ROUND}(3.2) = 3$

❑ Hàm **TRUNC(Num)**

- Num có kiểu Real.

Ý nghĩa giống như hàm INT. Khi cần kiểu trả về là Integer, bạn phải dùng hàm TRUNC(Num).

Ví dụ: $\text{TRUNC}(4.6) = 4$

❑ Hàm **ODD(Num)**

- Num có kiểu là Integer.
- Hàm này có kết quả là True nếu Num là số nguyên lẻ, trái lại sẽ cho kết quả là False. Vậy kiểu kết quả là Boolean.

Ví dụ: $\text{ODD}(4) = \text{FALSE}$

❑ Hàm **RANDOM**

- Kiểu kết quả là Real.
- Kết quả là số nguyên ngẫu nhiên $0 \leq n \leq \text{Num}$

❑ Hàm **UPCASE(Ch)**

- Ch có kiểu Char.
- Kết quả là kí tự hoa tương ứng với Ch, kiểu kết quả là Char.

Ví dụ: $\text{UPCASE}('h') = H$

❑ Hàm **COPY(Ch, Vitri, So)**

- Ch là biểu thức kiểu String.
- Vitri và So là biểu thức kiểu Integer.
- Hàm này trả về một chuỗi gồm có So kí tự, bắt đầu từ Vitri trong chuỗi Ch.

Ví dụ: Nếu chuỗi Ch = 'PASCAL', $\text{COPY}(\text{Ch}, 4, 3)$ sẽ là CAL

❑ Hàm **LENGTH(Ch)**

- Ch là một chuỗi
- Cho biết chiều dài của chuỗi Ch, kiểu kết quả là Integer.

Ví dụ: $\text{LENGTH}('PASCAL') = 6$

❑ Hàm **POS(SubCh, Ch)**

- SubCh, ch là chuỗi
- Hàm này cho ta biết vị trí xuất hiện đầu tiên của SubCh ở trong biểu thức Ch, nếu SubCh không nằm trong Ch thì nó sẽ cho trị 0.

Ví dụ: nếu Ch = 'PASCAL', thì POS('AS', Ch) = 2, POS('L', Ch) = 6
POS('T', Ch) = 0

2/ Thủ tục

❑ **CLRSCR**: Xoá màn hình và đưa dấu nháy về dòng 1 và cột 1 của màn hình.

❑ **GOTOXY(X, Y)**: Đem dấu nháy về dòng Y cột X

❑ **EXIT**: Nếu Exit thuộc chương trình con thì chấm dứt chương trình con và trở về chỗ gọi nó. Nếu thuộc chương trình chính thì sẽ chấm dứt chương trình.

❑ **HALT**: Chấm dứt thực hiện chương trình.

❑ **DELAY(time)**

▪ Time có kiểu Integer, tính theo 1/1000 giây.

▪ Thủ tục DELAY(time) dừng một thời gian là time.

❑ **DELETE(Ch, Vitri, So)**

▪ Ch có kiểu chuỗi

▪ Vitri, So kiểu Integer

▪ Thủ tục này sẽ xoá trong biểu thức Ch một số kí tự là So, bắt đầu từ vị trí.

Ví dụ: Nếu Ch = 'PASCALVISUALBASIC'

thì DELETE(Ch, 7, 11) sẽ được PASCAL

❑ **INSERT(Ch1, Ch2, Vitri)**

▪ Ch1, Ch2 là biểu thức kiểu chuỗi.

▪ Vitri là biểu thức kiểu Integer.

▪ Thủ tục này sẽ thêm Ch1 vào trong chuỗi Ch2 ở trước vị trí Vitri

Ví dụ: nếu Ch2='ASCAL' thì Insert('P', Ch2, 1) ta sẽ được 'PASCAL'.

Nếu Vitri vượt quá chiều dài của chuỗi Ch2 thì Ch1 sẽ được nối vào sau chuỗi Ch2. Ví dụ: Insert('BASIC', Ch2, 8) sẽ được 'ASCALBASIC'

❑ **STR(Giatri, Ch)**

▪ Giatri là một biểu thức Integer hoặc Real có ghi dạng in ra

▪ Ch là biến kiểu chuỗi.

▪ Thủ tục này cho chúng ta Ch là dạng chuỗi biểu diễn cho Giatri.

Ví dụ: j := 12345; Thì STR(j:6, Ch) cho ta Ch=' 12345' (có 1 khoảng trắng ở trước). Nếu dùng STR(j, Ch) thì ta được '12345'.

❑ **VAL(Ch, Bien, Loi)**

▪ Ch là biểu thức kiểu chuỗi.

▪ Bien là biến có kiểu là Integer hoặc Real.

▪ Loi là biến có kiểu là Integer.

▪ Thủ tục này sẽ biến chuỗi Ch thành số và gán vào Bien (Ch phải là là một chuỗi biểu diễn số nguyên hoặc số thực).

Ví dụ: Ch ='2004' thì VAL(Ch, x, e) sẽ cho x=2004, e=0.

Nếu Ch='2004A' thì VAL(Ch, x, e) sẽ cho x không xác định, e=5 (kí tự thứ 5 trong chuỗi Ch bị lỗi).


▪ Nếu không có lỗi, e=0, nếu có lỗi, e sẽ bằng vị trí đầu tiên gây ra lỗi.

▪ Trong chuỗi Ch không được có dấu trắng ' đi trước hoặc sau số.



TÓM TẮT

📖 Cần nắm vững các dữ liệu kiểu số, kiểu Logic, kiểu Char, kiểu String, cách khai báo các kiểu, biết giá trị giới hạn của từng loại kiểu, khi khai báo nên chọn kiểu gì cho phù hợp, tránh tốn nhiều bộ nhớ, biết cách khai báo các biến, các toán tử sử dụng trong mỗi kiểu.

 Cần biết thêm các hàm và thủ tục dùng trong các kiểu và cách dùng hàm và thủ tục trong chương trình để thực hiện những yêu cầu của đầu bài một cách nhanh chóng.

BÀI TẬP THỰC HÀNH

- 1 Xác định các kiểu dữ liệu và giá trị giới hạn cho từng biến qua các khai báo biến như sau:
A/ x : Integer;
B/ x : Byte;
C/ y : Longint;
D/ Chuoi : Char;
E/ Chuoi : String;
F/ Ketqua : Real;
G/ Tongso : Double;
H/ Ketqua : Boolean;
I/ Ketqua : Single;
- 2 Tìm câu sai trong các khai báo biến sau đây:
A/ Tongso : Real;
B/ Soam : Int;
C/ Soduong : Integer;
D/ Dungsai : Boolean;
- 3 Thêm, bớt sửa lại chương trình sau cho đúng (có ghi chú cho bạn chỗ sai, ký hiệu ‘:=’ là phép gán trong Pascal). Kiểm tra lại bằng cách chạy thử chương trình:
Program TIMCHOSAI {thiếu}
Var
m : Integer;

```

n : Real;
ch : Chuoi; {sai}
Begin
  m := 19;
  n := 25.62;
  Ch := 'Ban thay hoc Pascal co kho lam không?'; {thiếu}
  a := 'Hoc di doi voi hanh'; {sai, vì sao?}
  Writeln ('So nguyen m =', m);
  Writeln ('So thuc n =', n);
  Writeln ('Hoi ban mot chut', Ch); {thiếu}
  Readln;
End.

```

- ④ Tìm chỗ sai và thiếu trong các chương trình sau, nêu rõ điều sai hoặc thiếu:

```

Program BongdaSEAGAMES22;
X : Integer;
Ch1 : Ctring;
Ch2 : String;
Begin
  Ch1 := 'Doi tuyen bong da VIETNAM'
  Ch2 := 'Doi tuyen bong da THAILAN'
  X := 1;
  Writeln('Trong tran dau ngay 30/11/2003 vao luc 17h30 ', Ch1, ' Da
    hoa ', Ch2, ' voi ti so ', x, '-', x);
  Writeln('Ban cam thay vui hay nuoi tiec cho doi tuyen Viet Nam');
  Readln;
End.

```

Ghi chú: Cho biết chỗ sai trong chương trình tính từ trên xuống: **thiếu, sai, thiếu, thiếu.**

- ⑤ Nhập vào chương trình như sau, chạy xem kết quả và nhận xét.

```

Program THUHAM;
Var
X : Real;
Y : Real;
Z : Integer;
Ch1 : String;
Ch2 : String;
Begin
  X := 15.56;
  Y := 20.92;
  Z := 5;
  Ch1 := 'Seagame2';
  Ch2 := 'DuoctochuctaiVietNam';
  Writeln ('Gia tri tuyet doi cua X la: ', ABS(X));
  Writeln ('e luy thua Z la: ', EXP(Z));
  Writeln ('Logarith cua Z la: ', Ln(Z));
  Writeln ('Binh phuong cua Z la: ', SQR(Z));
  Writeln ('Can bac hai cua Z la: ', SQRT(Z));
  Writeln ('Phan nguyen cua X va Y la: ', INT(X), INT(Y));
  Writeln ('Phan le cua so X va Y la: ', FRAC(X), FRAC(Y));
  Writeln ('Ket qua tron so cua X va Y la: ', ROUND(X), ' ', ROUND(Y));

```

```
Writeln ('Phan nguyen cua X la "so nguyen": ', TRUNC(X));
Writeln ('Copy(Ch1,4,5) la: ', COPY(Ch1,4,5));
Writeln ('Chieu dai cua chuoai Ch1 va Ch2 la: ', LENGTH(Ch1), ' ',
        LENGTH(Ch2));
Writeln ('Chu e co dau tien trong chuoai Ch1 la: ', POS('e', Ch1));
Readln;
End.
```

- ⑥ Viết chương trình có dùng CLRSCR như sau, bạn chạy thử chương trình 3 lần, nhận xét kết quả xuất hiện trên màn hình.

```
Program DungCLRSCR;
Uses CRT;
Begin
    Clrscr;
    Writeln ('Chao mung Sea Games 22 to chuc tai Viet Nam');
    Readln;
End.
```

Câu lệnh đơn

KIẾN THỨC YÊU CẦU

- Biết cách sử dụng sắp xếp câu lệnh trong chương trình.
- Biết cách sử dụng các lệnh, thủ tục, hàm ... của các đơn vị chuẩn, biết cách khai báo bằng lệnh Uses.
- Biết ý nghĩa và khai báo các kiểu Type, const và khai báo các biến.
- Biết sơ qua về hàm và thủ tục, cách khai báo và cách gọi trong thân của chương trình.
- Biết dùng các câu lệnh và các phát biểu có thể dùng trong thân của chương trình.
- Biết các lệnh nhập dữ liệu từ bàn phím, lệnh xuất dữ liệu ra màn hình và máy in.

I/ Sử dụng câu lệnh của Pascal

1/ Câu lệnh khai báo dữ liệu:

Sau khai báo Program tên chương trình dấu chấm phẩy là khai báo đơn vị chuẩn (Unit) nếu như bạn dùng lệnh, hàm, thủ tục ... liên quan đến đơn vị chuẩn đó. Bạn khai báo thư viện lệnh (Uses) như sau:

Khai báo: **Uses tên đơn vị;**

Ví dụ: Trong bài tập 6 của bài trước, khi dùng lệnh **CLRSCR**; để xóa thông tin trên màn hình, đưa dấu nháy về góc trái trên của màn hình. Lệnh này thuộc đơn vị chuẩn **CRT**, nếu bạn không khai báo **Uses Crt**; trình biên dịch sẽ báo lỗi vì nó không hề biết lệnh Clrscr, nó xem như bạn chưa định nghĩa biến này.

Turbo Pascal có các đơn vị chuẩn như : **Crt, Dos, Graph, Gràph, Overlay, Printer, System, Turbo** và **Windos**.

Khai báo kiểu **Type** thường để khai báo một cấu trúc dùng trong chương trình.

Ví dụ: khai báo một kiểu tập hợp

Type

TapN = Set of Integer;
Traicay = (Nhan, saurieng, cam, quyt);
Taptraicay = Set of Traicay;

Ví dụ: Khi khai báo một mẫu tin

Type

Hocsinh = record
Hoten : String[40];
Namsinh : 1990 .. 1995;

Noisinh : String;
End;
Khai báo **Const** dùng để khai báo một hằng nào đó.

Ví dụ:

Const

n = 10;

Khai báo **biến** thường là khai báo sau cùng, trước các **hàm** và **thủ tục** (nếu có), trước **thân của chương trình**. Trước khi khai báo biến bạn phải dùng từ khoá **VAR** như đã đề cập ở trên.

Ví dụ:

Var

x,y,n,m : Integer;

k : Real;

Ketqua : String;

2/ Sử dụng hàm và chương trình

Để cho một chương trình sáng sủa dễ hiểu, người ta thường dùng **hàm** và **thủ tục** trong chương trình, vị trí của chúng thường được đặt trước thân của chương trình chính.

a/ hàm

Function tênhàm(các thông số (cách nhau bằng dấu ‘;’): kiểu trả về;

Var

.....

.....

Begin

.....

.....

End;

b/ Thủ tục

Procedure tênthủtục(các thông số (cách nhau bằng dấu ‘;’);

Var

.....

.....

Begin

.....

.....

End;

Ghi chú: Hàm và thủ tục các bạn sẽ được học trong các bài sau

3/ Các câu lệnh dùng trong thân chương trình

▪ *Câu lệnh đơn:* như **lệnh gán** giá trị (:=). Ví dụ: x:=5; y:=8; z:=7.9;

Ch1:= ‘Doi bong da Viet Nam’; ...**Lệnh gọi thủ tục, gọi hàm**

ví dụ: Có một thủ tục

Procedure Nhap(Var an:Mang; m,n:Integer; x,y:Integer);

Var

.....

Begin

.....

.....

End;

Begin {Thân chương trình}

.....

.....

Nhap(a,m,n,WhereX,WhereY); {Lệnh gọi thủ tục}

.....
.....
End.

Giả sử bạn có một hàm như sau:
Function Max(m:Real; n:Real;):Real;
Begin

.....
.....
End;
Begin

.....
.....
WriteLn('So lon nhat cua hai so tren la: ', Max(a,b); {lệnh gọi hàm})
ReadLn;
End.

- Các câu lệnh nhập, xuất:

- Read(biến1, biến2, ..., biến n); Write(mục1, mục2, ..., mục n);**

- Các câu lệnh có cấu trúc: Như lệnh lựa chọn **If**, **case**, lệnh lặp như **For**, **While**, **Repeat**. Các lệnh này các bạn sẽ được học trong các bài sắp tới.

- Câu lệnh ghép: **Begin end;**

Ghi chú: Sau End của hàm, thủ tục, câu lệnh ghép là dấu chấm phẩy ';'.

III/ Lệnh nhập - xuất dữ liệu

1/ Lệnh nhập dữ liệu

Từ bàn phím: Để nhập dữ liệu từ bàn phím vào cho các biến của chương trình, bạn dùng lệnh **Read** hoặc **ReadLn** như sau:

- Read(biến1, biến2, ..., biếnN);**

- ReadLn(biến1, biến2, ..., biếnN);**

- Trong đó các biến1, biến2, ..., biếnN phải có kiểu số nguyên, số thực, ký tự hoặc kiểu chuỗi.

- Mỗi câu lệnh Read hoặc ReadLn đều dừng chương trình để cho chúng ta nhập dữ liệu vào các biến, nếu nhập chưa đủ, máy chờ cho đến khi nhập xong.

- Lệnh **ReadLn;**

- Không đọc giá trị gì cả, chờ cho đến khi nhấn phím **Enter** sẽ trở về màn hình soạn thảo, được dùng để kiểm tra kết quả của chương trình, nếu không có lệnh này, khi nhấn phím Ctrl+F9 bạn không thấy kết quả, nó sẽ về ngay màn hình soạn thảo, muốn thấy, bạn phải nhấn **Alt+F5**.

- Lệnh gán giá trị cho một biến

Khi muốn gán giá trị cho một biến đã được khai báo, bạn dùng dấu hai chấm và dấu bằng (:=).

Ví dụ: x := 25; y:= 3.6; Chuoi := 'Hoc Pascal khong co gi kho';

2/ Lệnh xuất dữ liệu

Để xuất dữ liệu ra màn hình, chúng ta dùng lệnh Write hoặc WriteLn.

- Write(mục1, mục2, ..., mụcN);**

- Lệnh này viết ra các mục từ mục1 đến mụcN, không xuống hàng sau khi viết xong mục cuối cùng. Các mục có thể là biến, hằng, biểu thức ... nếu là chuỗi thì có bao hai dấu nháy đơn, nếu là biến, chỉ cần ghi tên biến.

Ví dụ:

Program LenhWrite;

Begin

- Write('Chuc mung ban');

- Write('den voi ngon ngu Pascal.');

- ReadLn;

End.

Chạy thử chương trình, sẽ có kết quả hiện ra như sau:

Chuc mung ban den voi ngon ngu Pascal.

Mặc dầu dùng hai câu lệnh Write, nhưng kết quả xuất hiện trên một dòng như kết quả của chương trình trên, bạn có thể dùng:

```
Program LenhWrite;
```

```
Begin
```

```
Write('Chuc mung ban ', 'den voi ngon ngu Pascal.');
```

```
Readln;
```

```
End.
```

Bạn có thể xem ví dụ sau:

```
Program LenhWrite;
```

```
Var
```

```
X : integer;
```

```
Begin
```

```
X:= 14;
```

```
Write('Ban Son nam nay duoc: ', X, ' tuoi.');
```

```
Readln;
```

```
End.
```

Kết quả sẽ là **Ban Son nam nay duoc 14 tuoi.**

Lệnh **Writeln(mục1, mục2, ..., mụcN);**

▪ Lệnh này sẽ xuống dòng sau khi viết mục cuối cùng, các mục cũng có ý nghĩa như trong lệnh Write.

Ví dụ:

```
Program LenhWriteln;
```

```
Begin
```

```
Writeln('Chuc mung ban');
```

```
Writeln('Den voi ngon ngu Pascal.');
```

```
Readln;
```

```
End.
```

Kết quả sẽ là:

Chuc mung ban

Den voi ngon ngu Pascal.

```
Program LenhWriteln;
```

```
Begin
```

```
Writeln('Chao mung Sea Games 22');
```

```
Write('Duoc to chuc tai ');
```

```
Write('Viet Nam.');
```

```
End.
```

Kết quả sẽ là:

Chào mung Sea Games 22

Duoc to chuc tai Viet Nam.

Write(R:10:6); sẽ viết số thực chiếm 10 chỗ, trong đó 6 chỗ dành cho phần thập phân.

Write(I:10); sẽ viết số nguyên chiếm 10 chỗ.

Để xuất dữ liệu ra máy in, bạn phải khai báo trong chương trình:

Uses Printer;

Dùng câu lệnh Write hoặc Writeln có dạng như sau:

Write(LST, mục1, mục2, ..., mụcN);

Writeln(LST, mục1, mục2, ..., mụcN);

▪ Với LST là tên thiết bị máy in đã khai báo trong Unit Printer.



TÓM TẮT

- ✎ Khi dùng các lệnh, hàm, thủ tục trong các đơn vị chuẩn, bạn phải khai báo đúng. Ví dụ: nếu dùng lệnh `Clrscr`, `Gotoxy(x,y)` bạn phải khai báo `Uses Crt`; khi muốn xuất dữ liệu ra màn hình, bạn phải khai báo `Uses Printer`; ...
- ✎ Dùng khai báo `Type` để khai báo các cấu trúc như tập hợp, mẫu tin ... nếu có dùng hằng số thì khai báo `Const`. Trước khi khai báo các biến nhớ có từ khoá `Var`.
- ✎ Cấu trúc của một hàm bắt đầu bằng `Function`, một thủ tục bắt đầu bằng `Procedure`. Xem lại cách gọi chúng trong chương trình.
- ✎ Lệnh gán dùng `:=`, khi phải lựa chọn một vấn đề dùng phát biểu `If` hoặc `Case`, khi cần công việc lặp đi lặp lại, dùng phát biểu `For`, `While` hoặc `Repeat`. Câu lệnh ghép sẽ bắt đầu bằng từ khoá `Begin` và kết thúc bằng `End` và dấu chấm phẩy.
- ✎ Khi nhập dữ liệu từ bàn phím dùng lệnh `Read` hoặc `Readln`.
- ✎ Khi cần xuất dữ liệu ra màn hình dùng lệnh `Write` hoặc `Writeln`, nếu dùng `Write` thì không xuống hàng, nếu dùng `Writeln` thì sẽ xuống hàng sau khi thực hiện. Nếu dùng khai báo `Uses Printer`; sẽ xuất ra máy in.



BÀI TẬP THỰC HÀNH

- ❶ Viết một chương trình xuất ra màn hình hai chuỗi: 'Chao mừng doi tuyen bong da Viet Nam' và 'Tai Sea Game 22'.
A/ Dùng một lệnh Write để kết quả hai chuỗi hiện ra trên một dòng.
B/ Dùng hai lệnh Write để có kết quả như câu A.
C/ Dùng hai lệnh để chuỗi một sẽ xuất hiện ở dòng trên, chuỗi hai xuất hiện ở dòng sau.
D/ Dùng thêm lệnh Clrscr. giải thích kết quả hiện ra trên màn hình khi thực hiện lệnh này.
- ❷ Tìm câu đúng trong các lệnh gán các biến sau đây:
A/ Tongso := Real;
B/ Soam =: Int;
C/ Soduong :- Integer;
D/ Dungsai :: Boolean;
- ❸ Thêm, bớt sửa lại chương trình sau cho đúng, kiểm tra lại bằng cách chạy thử chương trình:
Program TIMCHOSAI;
Var
m : Integer;
n : Real;
ch : String;
Begin
Clrscr;
m =: 19;
n := 25.62;

```

Ch := 'Hoc di doi voi hanh';
Writeln ('So nguyen m =', 'm');
Writeln ('So thuc n =', n);
Writeln 'Hoi ban mot chut', 'Ch');
Readln;

```

End.

- 4) Viết chương trình có khai báo 3 biến x, y và z có kiểu nguyên, thực hiện phép gán x bằng 2, y bằng 4 và z bằng tích của hai số x và y. Xuất kết quả ra màn hình câu:

A/ Tong hai so x va y la: (kết quả đúng).

B/ Tong hai so x va y la:
(kết quả đúng)

C/ (Kết quả đúng) la ket qua cua tích hai so x va y.

- 5) Viết chương trình với khai báo 3 biến như trên, nhưng không dùng phép gán, mà nhập từ bàn phím hai biến x và y. sau đó báo ra kết quả giống như các trường hợp A, B và C ở trên.

- 6) Giải thích chương trình sau thực hiện điều gì.

```

Program TinhToan;
Uses Crt;
Var
Bien : real;
Ketquamot, Ketquahai, ketquaba : Real;
Begin
Clrscr;
Writeln('Nhap vao mot bien :');
Readln(Bien);
Ketquamot := Bien * Bien;
Ketquahai := 4 * Bien;
Ketquaba := Bien * SQRT(2);
Writeln('Do ban day la gi? = ', Ketquamot : 10 : 1);
Writeln('Ket qua nay la gi? = ', Ketquahai : 10 : 1);
Writeln('Cung cau hoi nhu tren ', Ketquaba : 10 : 2);
Readln;
End.

```

- 7) Viết chương trình tính diện tích hình tam giác theo công thức Hê Rông $S = \sqrt{p(p-a)(p-b)(p-c)}$ với $p = (a+b+c)/2$.
- 8) Viết chương trình tính 4 phép toán cơ bản cộng, trừ, nhân, chia hai số, với hai số được nhập từ bàn phím.
- 9) Viết chương trình tính số dư của phép chia số nguyên thứ nhất cho số nguyên thứ hai, với hai số nguyên được nhập từ bàn phím.