

BỘ NÔNG NGHIỆP VÀ PHÁT TRIỂN NÔNG THÔN
TRƯỜNG CAO ĐẲNG CƠ ĐIỆN XÂY DỰNG VIỆT XÔ
KHOA CÔNG NGHỆ THÔNG TIN VÀ NGOẠI NGỮ

GIÁO TRÌNH
MÔN HỌC: LẬP TRÌNH CĂN BẢN
NGHỀ: QUẢN TRỊ MẠNG
TRÌNH ĐỘ: TRUNG CẤP

TUYÊN BỐ BẢN QUYỀN:

Tài liệu này thuộc loại sách giáo trình nên các nguồn thông tin có thể được phép dùng nguyên bản hoặc trích dùng cho các mục đích về đào tạo và tham khảo.

Mọi mục đích khác mang tính lệch lạc hoặc sử dụng với mục đích kinh doanh thiếu lành mạnh sẽ bị nghiêm cấm.

MÃ TÀI LIỆU: MH13

TaiLieu.vn

LỜI GIỚI THIỆU

Ngày nay, khoa học máy tính thâm nhập vào mọi lĩnh vực. Tự động hóa hiện đang là ngành chủ chốt điều hướng sự phát triển thế giới. Bất cứ ngành nghề nào cũng cần phải hiểu biết ít nhiều về Công nghệ Thông tin và lập trình nói chung. Cụ thể, C là một ngôn ngữ lập trình cấp cao mà mọi lập trình viên cần phải biết. Vì thế, trong giáo trình này, chúng ta sẽ nghiên cứu chi tiết cấu trúc ngôn ngữ C.

Môn học này là nền tảng để tiếp thu hầu hết các môn học khác trong chương trình đào tạo. Mặt khác, nắm vững ngôn ngữ C là cơ sở để phát triển các ứng dụng.

Học xong môn này, sinh viên phải nắm được các vấn đề sau:

- Khái niệm về ngôn ngữ lập trình.
- Khái niệm về kiểu dữ liệu
- Kiểu dữ liệu có cấu trúc (cấu trúc dữ liệu).
- Khái niệm về giải thuật
- Ngôn ngữ biểu diễn giải thuật.
- Ngôn ngữ sơ đồ (lưu đồ), sử dụng lưu đồ để biểu diễn các giải thuật.
- Tổng quan về Ngôn ngữ lập trình C.
- Các kiểu dữ liệu trong C.
- Các lệnh có cấu trúc.
- Cách thiết kế và sử dụng các hàm trong C.
- Một số cấu trúc dữ liệu trong C.

Ninh bình, ngày .. tháng ... năm 2018

Tham gia biên soạn

1. Chủ biên ThS. Nguyễn Bá Quân

MỤC LỤC

LỜI GIỚI THIỆU	3
GIÁO TRÌNH MÔN HỌC: LẬP TRÌNH CĂN BẢN	6
CHƯƠNG 1.....	8
GIỚI THIỆU VỀ NGÔN NGỮ C	8
1. Giới thiệu về ngôn ngữ lập trình C	8
2. Khởi động và thoát khỏi chương trình	9
2.1. Khởi động chương trình C	9
2.2. Thoát khỏi trường trình C	10
2.3. Các ví dụ đơn giản	10
CHƯƠNG 2.....	12
CÁC THÀNH PHẦN CƠ BẢN TRONG NGÔN NGỮ C.....	12
1. Từ khóa	12
2. Tên.....	13
3. Kiểu dữ liệu.....	13
3.1. Kiểu số nguyên.....	13
3.2. Kiểu số thực	13
3.3. Các phép toán.....	14
4. Ghi chú	16
5. Khai báo biến	17
6. Nhập/xuất dữ liệu.....	18
CHƯƠNG 3.....	21
CẤU TRÚC Rẽ NHÁNH CÓ ĐIỀU KIỆN	21
1. Lệnh và khối lệnh.....	21
1.1. Lệnh.....	21
1.2. Khối lệnh.....	21
2. Lệnh if	21
2.1. Dạng 1 (if thiếu).....	22
2.2. Dạng 2 (if đủ)	22
2.3. Cấu trúc else if	23
2.4. Cấu trúc if lồng nhau.....	25
3. Lệnh switch().....	26
3.1. Cấu trúc switch...case (switch thiếu)	26
3.2. Cấu trúc switch...case...default (switch đủ)	28
Bài tập thực hành	30
3.3. Cấu trúc switch lồng	31
CHƯƠNG 4.....	33
CẤU TRÚC VÒNG LẶP.....	33
1. Lệnh for.....	33
2. Lệnh break.....	35
3. Lệnh continue.....	36
4. Lệnh while.....	37
5. Lệnh do..while.....	38
6. Vòng lặp lồng nhau	39
CHƯƠNG 5.....	41

HÀM	41
1. Các ví dụ về hàm.....	41
1.1. Khái niệm về hàm	41
1.2. Các ví dụ về hàm.....	43
2. Tham số dạng tham biến tham trị.	45
2.1. Tham số dạng tham trị.	45
2.2. Tham số dạng tham biến	47
CHƯƠNG 6.....	54
MẢNG	54
GIỚI THIỆU KIỂU DỮ LIỆU “KIỂU MẢNG” TRONG C	54
1. Mảng một chiều.....	54
1.1. Khai báo mảng.....	55
1.2. Các thao tác tên mảng 1 chiều	55
1.3. Nhập xuất trong mảng 1 chiều.	58
1.4. Các bài tập cơ bản trên mảng 1 chiều	59
2. Mảng hai chiều.....	60
2.1. Cách khai báo, khởi tạo mảng 2 chiều	60
2.2. Nhập xuất mảng hai chiều.....	61
2.3. Bài tập thực hành	63
TÀI LIỆU THAM KHẢO	65

GIÁO TRÌNH MÔN HỌC: LẬP TRÌNH CĂN BẢN

Mã số môn học: MH 13

Vị trí, tính chất của môn học:

- Vị trí: Môn học được bố trí sau khi học sinh học xong các môn học chung, các môn học cơ sở chuyên ngành đào tạo chuyên môn nghề.

- Tính chất: Là môn học cơ sở nghề.

Mục tiêu của môn học:

- Kiến thức:
 - + Trình bày được công dụng của ngôn ngữ lập trình C⁺⁺;
 - + Trình bày được cú pháp, công dụng của các câu lệnh trong ngôn ngữ C⁺⁺;
 - + Phân tích được chương trình: xác định nhiệm vụ chương trình;
- Kỹ năng: Viết được chương trình và thực hiện biên dịch chương trình đơn giản trong máy tính bằng ngôn ngữ C⁺⁺;
- Năng lực tự chủ và trách nhiệm:
 - + Cần cù, chủ động trong học tập, đảm bảo an toàn trong học tập;
 - + Thể hiện thái độ tích cực, ham học hỏi và yêu quý ngành nghề đang học.

Nội dung môn học:

Nội dung tổng quát và phân bổ thời gian:

Số TT	Tên chương, mục	Thời gian			
		Tổng số	Lý thuyết	Thực hành, thí nghiệm, thảo luận, Bài tập	Kiểm tra
1	Chương 1: Giới thiệu về ngôn ngữ C⁺⁺ 1. Giới thiệu 2. Khởi động và thoát chương trình	1 0.5 0.5	1 0.5 0.5		
2	Chương 2: Các thành phần trong ngôn ngữ C⁺⁺ 1. Từ khóa 2. Tên 3. Kiểu dữ liệu 4. Ghi chú 5. Khai báo biến 6. Nhập/xuất dữ liệu	7 1 1 1 1 2	6 1 1 1 1 1	1 1	
3	Chương 3: Cấu trúc rẽ nhánh có điều kiện 1. Lệnh và khối lệnh 1.1. Lệnh 1.2. Khối lệnh 2. Lệnh if 2.1. Dạng 1 (if thiếu) 2.2. Dạng 2 (if đủ)	12 2 1 1 6 2 1	5 1 0.5 0.5 3 1 1	6 1 0.5 0.5 2 1	1 1

	2.3. Cấu trúc else if	2		1	1
	2.4. Cấu trúc if lồng nhau	1	1		
	3. Lệnh switch	4	2	2	
	3.1. Cấu trúc switch...case (switch thiếu)	2	1	1	
	3.2. Cấu trúc switch...case...default	1	0.5	0.5	
	3.3. Cấu trúc switch lồng nhau	1	0.5	0.5	
4	Chương 4: Cấu trúc vòng lặp	15	6	8	1
	1. Lệnh for	3.5	1.5	2	
	2. Lệnh Break	1	0.5	0.5	
	3. Lệnh continue	1	0.5	0.5	
	4. Lệnh while	2.5	1	1.5	
	5. Lệnh do...while	2.5	1	1.5	
	6. Vòng lặp lồng nhau	2	1	1	
	7. So sánh sự khác nhau của các vòng lặp	2.5	0.5	1	1
5	Chương 5: Hàm	10	4	5	1
	1. Các ví dụ về hàm	3	1	2	
	1.1. Khái niệm về hàm	1	1		
	1.2. Các ví dụ về hàm	2		2	
	2. Tham số dạng tham biến và tham trị	4	2	2	
	2.1 Tham số dạng tham biến	2	1	1	
	2.2. Tham số dạng tham trị	2	1	1	
	3. Sử dụng biến toàn cục	3	1	1	1
6	Chương 6: Mảng	15	7	7	1
	1. Mảng 1 chiều	9	4	4	1
	1.1. Cách khai báo mảng	1	1		
	1.2. Các thao tác tên mảng 1 chiều	2	1	1	
	1.3. Nhập xuất mảng 1 chiều	2	1	1	
	1.4. Các bài tập cơ bản trên mảng 1 chiều	4	1	2	1
	2. Mảng 2 chiều	6	3	3	
	2.1. Cách khai báo, khởi tạo mảng 2 chiều	1	1		
		2	1	1	
	2.2. Nhập xuất mảng 2 chiều	3		3	
	2.3. Các bài tập cơ bản trên mảng 2 chiều				
	Cộng	60	29	27	4

CHƯƠNG 1

GIỚI THIỆU VỀ NGÔN NGỮ C

Mã chương: MH13-01

Giới thiệu:

C là ngôn ngữ rất có hiệu quả và được ưa chuộng nhất để viết các phần mềm hệ thống, mặc dù nó cũng được dùng cho việc viết các ứng dụng. Ngoài ra, C cũng thường được dùng làm phương tiện giảng dạy trong khoa học máy tính mặc dù ngôn ngữ này không được thiết kế dành cho người nhập môn.

Mục tiêu:

- Trình bày được lịch sử phát triển của ngôn ngữ C;
- Mô tả được những ứng dụng thực tế của ngôn ngữ C.
- Một số thao tác cơ bản của trình soạn thảo C.
- Tiếp cận một số lệnh đơn giản thông qua các ví dụ
- Thực hiện các thao tác an toàn với máy tính.

Nội dung:

1. Giới thiệu về ngôn ngữ lập trình C

Mục tiêu:

- Trình bày được lịch sử phát triển của ngôn ngữ C;
- Mô tả được những ứng dụng thực tế của ngôn ngữ C.
- Mô tả những đặc điểm cơ bản của ngôn ngữ C

C là ngôn ngữ lập trình cấp cao, được sử dụng rất phổ biến để lập trình hệ thống cùng với Assembler và phát triển các ứng dụng.

Vào những năm cuối thập kỷ 60 đầu thập kỷ 70 của thế kỷ XX, Dennis Ritchie (làm việc tại phòng thí nghiệm Bell) đã phát triển ngôn ngữ lập trình C dựa trên ngôn ngữ BCPL (do Martin Richards đưa ra vào năm 1967) và ngôn ngữ B (do Ken Thompson phát triển từ ngôn ngữ BCPL vào năm 1970 khi viết hệ điều hành UNIX đầu tiên trên máy PDP-7) và được cài đặt lần đầu tiên trên hệ điều hành UNIX của máy DEC PDP-11. Năm 1978, Dennis Ritchie và B.W Kernighan đã cho xuất bản quyển “Ngôn ngữ lập trình C” và được phổ biến rộng rãi đến nay.

Lúc ban đầu, C được thiết kế nhằm lập trình trong môi trường của hệ điều hành Unix nhằm mục đích hỗ trợ cho các công việc lập trình phức tạp. Nhưng về sau, với những nhu cầu phát triển ngày một tăng của công việc lập trình, C đã vượt qua khuôn khổ của phòng thí nghiệm Bell và nhanh chóng hội nhập vào thế giới lập trình để rồi các công ty lập trình sử dụng một cách rộng rãi. Sau đó, các công ty sản xuất phần mềm lần lượt đưa ra các phiên bản hỗ trợ cho việc lập trình bằng ngôn ngữ C và chuẩn ANSI C cũng được khai sinh từ đó.

Ngôn ngữ lập trình C là một ngôn ngữ lập trình hệ thống rất mạnh và rất “mềm dẻo”, có một thư viện gồm rất nhiều các hàm (function) đã được tạo sẵn. Người lập trình có thể tận dụng các hàm này để giải quyết các bài toán mà không cần phải tạo mới. Hơn thế nữa, ngôn ngữ C hỗ trợ rất nhiều phép toán nên phù hợp cho việc giải quyết các bài toán kỹ thuật có nhiều công thức phức tạp. Ngoài ra, C cũng cho phép người lập trình tự định nghĩa thêm các kiểu dữ liệu trừu tượng khác. Tuy nhiên, điều mà người mới vừa học lập trình

C thường gặp “rắc rối” là “hơi khó hiểu” do sự “mềm dẻo” của C. Dù vậy, C được phổ biến khá rộng rãi và đã trở thành một công cụ lập trình khá mạnh, được sử dụng như là một ngôn ngữ lập trình chủ yếu trong việc xây dựng những phần mềm hiện nay.

Ngôn ngữ C có những đặc điểm cơ bản sau:

+ Tính cô đọng (compact): C chỉ có 32 từ khóa chuẩn và 40 toán tử chuẩn, nhưng hầu hết đều được biểu diễn bằng những chuỗi ký tự ngắn gọn.

+ Tính cấu trúc (structured): C có một tập hợp những chỉ thị của lập trình như cấu trúc lựa chọn, lặp... Từ đó các chương trình viết bằng C được tổ chức rõ ràng, dễ hiểu.

+ Tính tương thích (compatible): C có bộ tiền xử lý và một thư viện chuẩn vô cùng phong phú nên khi chuyển từ máy tính này sang máy tính khác các chương trình viết bằng C vẫn hoàn toàn tương thích.

+ Tính linh động (flexible): C là một ngôn ngữ rất uyển chuyển và cú pháp, chấp nhận nhiều cách thể hiện, có thể thu gọn kích thước của các mã lệnh làm chương trình chạy nhanh hơn.

+ Biên dịch (compile): C cho phép biên dịch nhiều tập tin chương trình riêng rẽ thành các tập tin đối tượng (object) và liên kết (link) các đối tượng đó lại với nhau thành một chương trình có thể thực thi được (executable) thống nhất.

2. Khởi động và thoát khỏi chương trình.

Mục tiêu:

- Một số thao tác cơ bản của trình soạn thảo C.
- Tiếp cận một số lệnh đơn giản thông qua các ví dụ
- Thực hiện các thao tác an toàn với máy tính.

2.1. Khởi động chương trình C

Nhập lệnh tại dấu nhắc DOS: gõ BC ↵ (Enter) (nếu đường dẫn đã được cài bằng lệnh path trong đó có chứa đường dẫn đến thư mục chứa tập tin BC.EXE). Nếu đường dẫn chưa được cài đặt ta tìm xem thư mục BORLAND C (hoặc TURBO C) nằm trong ổ đĩa nào. Sau đó ta gõ lệnh sau:

<ổ đĩa>:\BORLAND C\BIN\BC↵(Enter)

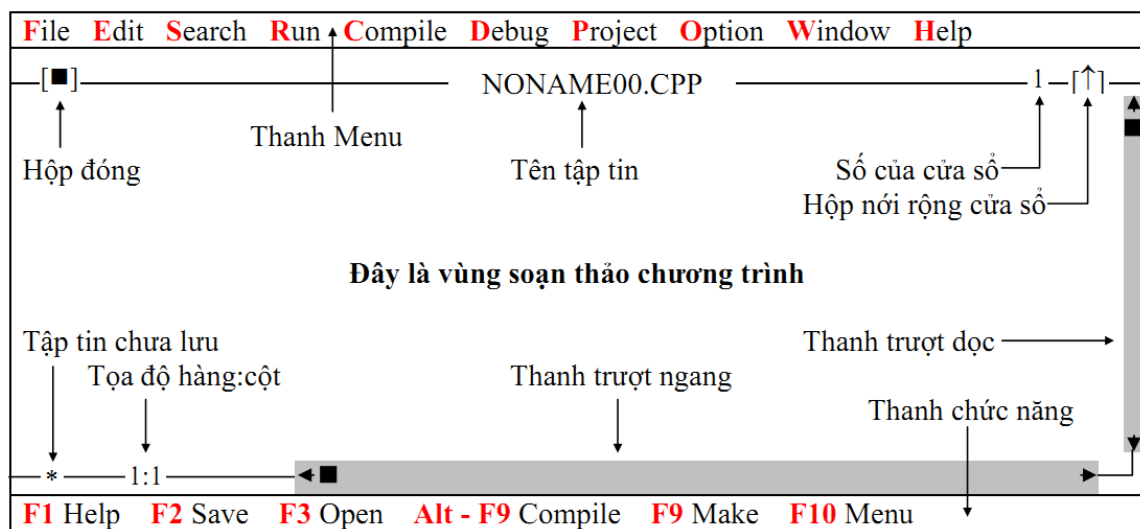
Nếu vừa khởi động BC vừa soạn thảo chương trình với một tập tin có tên do chúng ta đặt, thì gõ lệnh: BC [đường dẫn]<tên file cần soạn thảo>, nếu tên file cần soạn thảo đã có thì được nạp lên, nếu chưa có sẽ được tạo mới.

Khởi động tại Windows: Bạn vào menu Start, chọn Run, bạn gõ vào hộp Open 1 trong các dòng lệnh như nhập tại DOS. Hoặc bạn vào Window Explorer, chọn ổ đĩa chứa thư mục BORLANDC, vào thư mục BORLANDC, vào thư mục BIN, khởi động tập tin BC.EXE.

Ví dụ: gõ D:\BORLANDC\BIN\BC E:\BAITAP_BC\VIDU1.CPP

Câu lệnh trên có nghĩa khởi BC và nạp tập tin VIDU1.CPP chứa trong thư mục BAITAP_BC trong ổ đĩa E. Nếu tập tin này không có sẽ được tạo mới.

Màn hình sau khi khởi động thành công:



Hình 1.1. Màn hình sau khi khởi động C thành công

2.2. Thoát khỏi trường trình C

Ấn phím F10 (kích hoạt Menu), chọn menu File, chọn Quit;

Hoặc ấn tổ hợp phím Alt – X

2.3. Các ví dụ đơn giản

```
1  /* Chương trình in ra câu bài học C
   đầu tiên */
2  #include<stdio.h>
3
4  void main()
5  {
6      printf("Bai hoc C dau tien");
7  }
```

Kết quả in ra màn hình:

Bai hoc C dau tien

Dòng thứ 1: bắt đầu bằng /* và kết thúc bằng */ cho biết hàng này là hàng diễn giải (chú thích). Khi dịch và chạy chương trình, dòng này không được dịch và cũng không thi hành lệnh gì cả. Mục đích của việc ghi chú này giúp chương trình rõ ràng hơn. Sau này chúng ta đọc lại chương trình biết chương trình làm gì.

Dòng thứ 2: chứa phát biểu tiền xử lý #include <stdio.h>. Vì trong chương trình này chúng ta sử dụng hàm thư viện của C là printf, do đó chúng ta cần phải có khai báo của hàm thư viện này để báo cho trình biên dịch C biết. Nếu không khai báo chương trình sẽ báo lỗi.

Dòng thứ 3: Hàng trống viết ra với ý đồ làm cho bảng chương trình thoáng, dễ đọc.

Dòng thứ 4: void main(void) là thành phần chính của mọi chương trình C (bạn có thể viết main() hoặc void main() hoặc main(void)). Tuy nhiên, bạn nên viết theo dạng void main(void) để chương trình rõ ràng hơn. Mọi chương trình C đều bắt đầu thi hành từ hàm main. Cặp dấu ngoặc () cho biết đây là khối hàm (function). Hàm void main(void) có từ khóa void đầu tiên cho biết hàm này

không trả về giá trị, từ khóa void trong ngoặc đơn cho biết hàm này không nhận vào đối số.

Dòng thứ 5 và 7: cặp dấu ngoặc móc {} giới hạn thân của hàm. Thân hàm bắt đầu bằng dấu { và kết thúc bằng dấu }.

Dòng thứ 6: printf(“Bai hoc C dau tien.”);, chỉ thị cho máy in ra chuỗi ký tự nằm trong nháy kép (“”). Hàng này được gọi là một câu lệnh, kết thúc một câu lệnh trong C phải là dấu chấm phẩy (;).

Chú ý:

- Các từ include, stdio.h, void, main, print phải viết bằng chữ thường.
- Chuỗi trong nháy kép cần in ra “Bạn có thể viết chữ HOA, thường tùy ý”.
- Kết thúc câu lệnh phải có dấu chấm phẩy
- Kết thúc tên hàm không có dấu chấm phẩy hoặc bất cứ dấu gì.
- Ghi chú phải đặt trong cặp /*.....*/
- Thân hàm phải được bao bởi cặp {}.
- Các câu lệnh trong thân hàm phải viết thụt vào.

Sau khi nhập xong đoạn chương trình vào máy. Bạn ấn và giữ phím Ctrl, gõ F9 để dịch và chạy chương trình. Khi đó chúng ta thấy chương trình chớp rất nhanh và không thấy kết quả gì cả. Bạn Ấn và giữ phím Alt, gõ F5 để xem kết quả, khi xem xong, ấn phím bất kỳ để quay về màn hình soạn thảo chương trình.

CHƯƠNG 2

CÁC THÀNH PHẦN CƠ BẢN TRONG NGÔN NGỮ C

Mã chương: MH13-02

Giới thiệu:

Ngôn ngữ C hỗ trợ rất nhiều phép toán nên phù hợp cho việc giải quyết các bài toán kỹ thuật có nhiều công thức phức tạp. Ngoài ra, C cũng cho phép người lập trình tự định nghĩa thêm các kiểu dữ liệu trừu tượng khác.

Mục tiêu:

- Sử dụng được hệ thống kí hiệu và từ khóa
- Khai báo tên đúng
- Trình bày được các kiểu dữ liệu
- Khai báo biến đúng
- Thực hiện được việc nhập và xuất dữ liệu
- Thực hiện các thao tác an toàn với máy tính

1. Từ khóa

***Mục tiêu:** Hiểu và sử dụng đúng hệ thống ký hiệu và từ khóa.*

Bộ chữ viết trong C

Ngôn ngữ C được xây dựng trên bộ ký tự sau:

- 26 chữ cái hoa: A B C .. Z
- 26 chữ cái thường: a b c .. z
- 10 chữ số: 0 1 2 .. 9
- Các ký hiệu toán học: + - * / = ()
- Ký tự gạch nối: _
- Các ký tự khác: . , ; [] { } ! \ & % # \$...
- Dấu cách (space) dùng để tách các từ. Ví dụ chữ SINH VIEN có 9 ký tự, còn SINHVIEN chỉ có 8 ký tự.

* Chú ý:

Khi viết chương trình, ta không được sử dụng bất kỳ ký tự nào khác ngoài các ký tự trên.

Từ khóa

Từ khóa là những từ được sử dụng để khai báo các kiểu dữ liệu, để viết các toán tử và các câu lệnh. Một số từ khóa của TURBO C được liệt kê như sau:

asm	break	case	cdecl
char	const	continue	default
do	double	else	enum
extern	Far	float	for
goto	Huge	if	int
interrupt	Long	near	pascal
register	return	short	signed
sizeof	static	struct	switch
typedef	union	unsigned	void
volatile	while		

Chú ý:

- Không được dùng các từ khóa để đặt tên cho các hằng, biến, mảng, hàm ...

- Từ khoá phải được viết bằng chữ thường, ví dụ: viết từ khoá khai báo kiểu nguyên là `int` chứ không phải là `INT`.

2. Tên

Mục tiêu: Hiểu và khai báo tên theo đúng nguyên tắc.

Khái niệm tên rất quan trọng trong quá trình lập trình, nó không những thể hiện rõ ý nghĩa trong chương trình mà còn dùng để xác định các đại lượng khác nhau khi thực hiện chương trình.

Tên thường được đặt cho hằng, biến, mảng, con trỏ, nhãn... Chiều dài tối đa của tên là 32 ký tự.

Tên biến hợp lệ là một chuỗi ký tự liên tục gồm: Ký tự chữ, số và dấu gạch dưới. Ký tự đầu của tên phải là chữ hoặc dấu gạch dưới. Khi đặt tên không được đặt trùng với các từ khóa.

Ví dụ 1 :

Các tên đúng: `delta`, `a_1`, `Num_ODD`, `Case`

Các tên sai:

`3a_1` (ký tự đầu là số)

`num-odd` (sử dụng dấu gạch ngang)

`int` (đặt tên trùng với từ khóa)

`del ta` (có khoảng trắng)

`f(x)` (có dấu ngoặc tròn)

Lưu ý: Trong C, tên phân biệt chữ hoa, chữ thường

Ví dụ 2 : `number` khác `Number`

`case` khác `Case`

(`case` là từ khóa, do đó bạn đặt tên là `Case` vẫn đúng)

3. Kiểu dữ liệu

Mục tiêu: Hiểu và có thể sử dụng kiểu dữ liệu theo đúng mục đích của bài toán.

3.1. Kiểu số nguyên

Trong C cho phép sử dụng số nguyên kiểu `int`, số nguyên dài kiểu `long` và số nguyên không dấu kiểu `unsigned`. Kích cỡ và phạm vi biểu diễn của chúng được chỉ ra trong bảng dưới đây:

Kiểu	Phạm vi biểu diễn	Kích thước
<code>int</code>	-32768 đến 32767	2 byte
<code>unsigned int</code>	0 đến 65535	2 byte
<code>long</code>	-2147483648 đến 2147483647	4 byte
<code>unsigned long</code>	0 đến 4294967295	4 byte

Bảng 1.1. Kích cỡ và phạm vi biểu diễn kiểu dữ liệu số nguyên

3.2. Kiểu số thực

Trong C cho phép sử dụng số thực gồm 3 kiểu: `float`, `double` và `long double`. Kích cỡ và phạm vi biểu diễn của chúng được chỉ ra trong bảng dưới đây:

Kiểu	Phạm vi biểu diễn	Số chữ số có nghĩa	Kích thước
<code>Float</code>	3.4E-38 đến 3.4E+38	7 đến 8	4 byte
<code>Double</code>	1.7E-308 đến 1.7E+308	15 đến 16	8 byte

Kiểu	Phạm vi biểu diễn	Số chữ số có nghĩa	Kích thước
long double	3.4E-4932 đến 1.1E4932	17 đến 18	10 byte

Bảng 1.2. Kích cỡ và phạm vi biểu diễn kiểu số thực

Giải thích:

Máy tính có thể lưu trữ được các số kiểu float có giá trị tuyệt đối từ 3.4E-38 đến 3.4E+38. Các số có giá trị tuyệt đối nhỏ hơn 3.4E-38 được xem bằng 0. Phạm vi biểu diễn của số double được hiểu theo nghĩa tương tự.

3.3. Các phép toán

Mục tiêu : Hiểu và có thể thực hiện các phép tính toán trong chương trình.

Các phép toán số học

Phép toán	Ý nghĩa	Ví dụ
+	Phép cộng	a+b
-	Phép trừ	a-b
*	Phép nhân	a*b
/	Phép chia	a/b (Chia số nguyên sẽ chắt phần thập phân)
%	Phép lấy phần dư	a%b (Cho phần dư của phép chia a cho b)

Bảng 1.3. Các phép toán số học

Có phép toán một ngôi - ví dụ -(a+b) sẽ đảo giá trị của phép cộng (a+b).

Ví dụ:

$$11/3=3$$

$$11\%3=2$$

$$-(2+6)=-8$$

Các phép toán + và - có cùng thứ tự ưu tiên, có thứ tự ưu tiên nhỏ hơn các phép *, /, % và cả ba phép này lại có thứ tự ưu tiên nhỏ hơn phép trừ một ngôi.

Các phép toán số học được thực hiện từ trái sang phải. Số ưu tiên và khả năng kết hợp của phép toán được chỉ ra trong một mục sau này.

Các phép toán quan hệ và logic

Phép toán quan hệ và logic cho ta giá trị đúng (1) hoặc giá trị sai (0). Nói cách khác, khi các điều kiện nêu ra là đúng thì ta nhận được giá trị 1, trái lại ta nhận giá trị 0.

Các phép toán quan hệ:

Phép toán	Ý nghĩa	Ví dụ
>	So sánh lớn hơn	a>b 4>5 có giá trị 0
>=	So sánh lớn hơn hoặc bằng	a>=b 6>=2 có giá trị 1
<	So sánh nhỏ hơn	a<b 6<=7 có giá trị 1
<=	So sánh nhỏ hơn hoặc bằng	a<=b 8<=5 có giá trị 0
==	So sánh bằng nhau	a==b 6==6 có giá trị 1
!=	So sánh khác nhau	a!=b 9!=9 có giá trị 0

Bốn phép toán đầu có cùng số ưu tiên, hai phép sau có cùng số thứ tự ưu tiên nhưng thấp hơn số thứ tự của bốn phép đầu.

Các phép toán quan hệ có số thứ tự ưu tiên thấp hơn so với các phép toán số học, cho nên biểu thức: $i < n-1$ được hiểu là $i < (n-1)$.

Các phép toán logic

Trong C sử dụng ba phép toán logic:

Phép phủ định một ngôi (!)

A	!a
khác 0	0
bằng 0	1

Phép và (AND) && - Phép hoặc (OR) ||

a	B	a&&b	a b
khác 0	khác 0	1	1
khác 0	bằng 0	0	1
bằng 0	khác 0	0	1
bằng 0	bằng 0	0	0

Các phép quan hệ có số ưu tiên nhỏ hơn so với ! nhưng lớn hơn so với && và ||, vì vậy biểu thức sau:

$(a < b) \&\& (c > d)$

có thể viết lại thành:

$a < b \&\& c > d$

* Chú ý: Cả a và b có thể là nguyên hoặc thực.

Phép toán tăng giảm

C đưa ra hai phép toán một ngôi để tăng và giảm các biến (nguyên và thực). Toán tử tăng là ++ sẽ cộng 1 vào toán hạng của nó, toán tử giảm -- thì sẽ trừ toán hạng đi 1.

Ví dụ:

$n=5$

$++n$ Cho ta $n=6$

$--n$ Cho ta $n=4$

Ta có thể viết phép toán ++ và -- trước hoặc sau toán hạng như sau: $++n$, $n++$, $--n$, $n--$

Sự khác nhau của $++n$ và $n++$ ở chỗ: Trong phép $n++$ thì tăng sau khi giá trị của nó đã được sử dụng, còn trong phép $++n$ thì n được tăng trước khi sử dụng. Sự khác nhau giữa $n--$ và $--n$ cũng như vậy.

Ví dụ:

$n=5$

$x=++n$ Cho ta $x=6$ và $n=6$

$x=n++$ Cho ta $x=5$ và $n=6$

Thứ tự ưu tiên các phép toán

Các phép toán có độ ưu tiên khác nhau, điều này có ý nghĩa trong cùng một biểu thức sẽ có một số phép toán này được thực hiện trước một số phép toán khác.

Thứ tự ưu tiên của các phép toán được trình bày như sau:

TT	Phép toán	Trình tự kết hợp
----	-----------	------------------

TT	Phép toán	Trình tự kết hợp
1	() [] ->	Trái qua phải
2	! ~ & * - ++ -- (type) sizeof	Phải qua trái
3	* (phép nhân) / %	Trái qua phải
4	+ -	Trái qua phải
5	<< >>	Trái qua phải
6	< <= > >=	Trái qua phải
7	= = !=	Trái qua phải
8	&	Trái qua phải
9	^	Trái qua phải
10		Trái qua phải
11	&&	Trái qua phải
12		Trái qua phải
13	?:	Phải qua trái
14	= += -= *= /= %= <<= >>= &= ^= =	Phải qua trái
15	,	Trái qua phải

Bảng 1.4. Thứ tự ưu tiên của các phép toán

Chú thích:

- Các phép toán tên một dòng có cùng thứ tự ưu tiên, các phép toán ở hàng trên có số ưu tiên cao hơn các số ở hàng dưới.
- Đối với các phép toán cùng mức ưu tiên thì trình tự tính toán có thể từ trái qua phải hay ngược lại được chỉ ra trong cột trình tự kết hợp.

Ví dụ:

--px=(--px) (Phải qua trái)

8/4*6=(8/4)*6 (Trái qua phải)

Nên dùng các dấu ngoặc tròn để viết biểu thức một cách chính xác.

4. Ghi chú

Mục tiêu : Biết cách ghi chú khi viết chương trình.

Ghi chú

Trong khi lập trình cần phải ghi chú để giải thích các biến, hằng, thao tác xử lý giúp cho chương trình rõ ràng dễ hiểu, dễ nhớ, dễ sửa chữa và để người đọc dễ hiểu. Trong C có các ghi chú sau: // hoặc /* nội dung ghi chú */

Ví dụ:

```
void main()
{
    int a, b; //khai bao bien t kieu int
    a = 1;    //gan 1 cho a
    b = 3;    //gan 3 cho b
    /*  thuat toan tim so lon nhat la
        neu a lon hon b thi a lon nhat
        nguoc lai b lon nhat */
    if (a > b) printf("max: %d", a);
    else printf("max: %d", b);
}
```

```

        getch ( ) ;
    }

```

Khi biên dịch chương trình, C gặp cặp dấu ghi chú sẽ không dịch ra ngôn ngữ máy.

Tóm lại, đối với ghi chú dạng // dùng để ghi chú một hàng và dạng /* */ có thể ghi chú một hàng hoặc nhiều hàng.

Cấu trúc chương trình C

Một chương trình C bao gồm các phần như: Các chỉ thị tiền xử lý, khai báo biến ngoài, các hàm tự tạo, chương trình chính (hàm main).

Cấu trúc có thể như sau:

- Các chỉ thị tiền xử lý (*Preprocessor directives*)

```
#include <Tên tập tin thư viện>
```

```
#define ....
```

Định nghĩa kiểu dữ liệu (phần này không bắt buộc): dùng để đặt tên lại cho một kiểu dữ liệu nào đó để gọi nhớ hay đặt 1 kiểu dữ liệu cho riêng mình dựa trên các kiểu dữ liệu đã có.

Cú pháp: typedef <Tên kiểu cũ> <Tên kiểu mới>

Ví dụ: typedef int SoNguyen; // Kiểu SoNguyen là kiểu int

Khai báo các prototype (tên hàm, các tham số, kiểu kết quả trả về,... của các hàm sẽ cài đặt trong phần sau, phần này không bắt buộc): phần này chỉ là các khai báo đầu hàm, không phải là phần định nghĩa hàm.

Khai báo các biến ngoài (các biến toàn cục) phần này không bắt buộc: phần này khai báo các biến toàn cục được sử dụng trong cả chương trình.

- Chương trình chính (phần này bắt buộc phải có)

```
<Kiểu dữ liệu trả về> main()
```

```
{
```

Các khai báo cục bộ trong hàm main: Các khai báo này chỉ tồn tại trong hàm mà thôi, có thể là khai báo biến hay khai báo kiểu.

Các câu lệnh dùng để định nghĩa hàm main.

```
return <kết quả trả về>; // Hàm phải trả về kết quả
```

```
}
```

Cài đặt các hàm

```
<Kiểu dữ liệu trả về> function1( các tham số)
```

```
{
```

Các khai báo cục bộ trong hàm.

Các câu lệnh dùng để định nghĩa hàm.

```
return <kết quả trả về>;
```

```
}
```

```
...
```

Một chương trình C bắt đầu thực thi từ hàm main (thông thường là từ câu lệnh đầu tiên đến câu lệnh cuối cùng).

5. Khai báo biến

Mục tiêu : *Biết cách khai báo biến theo đúng nguyên tắc đặt tên.*

Biến.

Mỗi biến cần phải được khai báo trước khi đưa vào sử dụng. Việc khai báo biến được thực hiện theo mẫu sau:

Kiểu dữ liệu của biến tên biến ;

Ví dụ :

int a,b,c;	Khai báo ba biến int là a,b,c
long dai,mn;	Khai báo hai biến long là dai và mn
char kt1,kt2;	Khai báo hai biến ký tự là kt1 và kt2
float x,y;	Khai báo hai biến float là x và y
double canh;	Khai báo một biến double là canh

Biến kiểu int chỉ nhận được các giá trị kiểu int. Các biến khác cũng có ý nghĩa tương tự. Các biến kiểu char chỉ chứa được một ký tự. Để lưu trữ được một xâu ký tự cần sử dụng một mảng kiểu char.

Vị trí khai báo biến trong C

Khi lập trình, chúng ta phải nắm rõ phạm vi của biến. Nếu khai báo và sử dụng không đúng, không rõ ràng sẽ dẫn đến sai sót khó kiểm soát được, vì vậy cần phải xác định đúng vị trí, phạm vi sử dụng biến trước khi sử dụng biến.

Khai báo biến ngoài (biến toàn cục): Vị trí biến đặt bên ngoài tất cả các hàm, cấu trúc... Các biến này có ảnh hưởng đến toàn bộ chương trình. Chu trình sống của nó là bắt đầu chạy chương trình đến lúc kết thúc chương trình.

Khai báo biến trong (biến cục bộ): Vị trí biến đặt bên trong hàm, cấu trúc.... Chỉ ảnh hưởng nội bộ bên trong hàm, cấu trúc đó. Chu trình sống của nó bắt đầu từ lúc hàm, cấu trúc được gọi thực hiện đến lúc thực hiện xong.

Biểu thức

Biểu thức là một sự kết hợp giữa các phép toán và các toán hạng để diễn đạt một công thức toán học nào đó. Mỗi biểu thức có sẽ có một giá trị. Như vậy hằng, biến, phần tử mảng và hàm cũng được xem là biểu thức.

Trong C, ta có hai khái niệm về biểu thức:

Biểu thức gán.

Biểu thức điều kiện .

Biểu thức được phân loại theo kiểu giá trị: nguyên và thực. Trong các mệnh đề logic, biểu thức được phân thành đúng (giá trị khác 0) và sai (giá trị bằng 0).

Biểu thức thường được dùng trong:

Vế phải của câu lệnh gán.

Làm tham số thực sự của hàm.

Làm chỉ số.

Trong các toán tử của các cấu trúc điều khiển.

6. Nhập/xuất dữ liệu

Mục tiêu: *Biết thực hiện lệnh nhập dữ liệu và xuất dữ liệu ra màn hình.*

Lệnh gán.

Biểu thức gán là biểu thức có dạng:

$v=e$

Trong đó v là một biến (hay phần tử mảng), e là một biểu thức. Giá trị của biểu thức gán là giá trị của e, kiểu của nó là kiểu của v. Nếu đặt dấu ; vào sau biểu thức gán ta sẽ thu được phép toán gán có dạng:

$v=e;$

Biểu thức gán có thể sử dụng trong các phép toán và các câu lệnh như các biểu thức khác. Ví dụ như khi ta viết

$a=b=5;$

thì điều đó có nghĩa là gán giá trị của biểu thức $b=5$ cho biến a . Kết quả là $b=5$ và $a=5$.

Hoàn toàn tương tự như: $a=b=c=d=6;$ gán 6 cho cả a, b, c và d

Ví dụ :

$z=(y=2)*(x=6);$ {ở đây $*$ là phép toán nhân}

gán 2 cho y , 6 cho x và nhân hai biểu thức lại cho ta $z=12$.

Lệnh nhập

Hàm `scanf` là hàm đọc thông tin từ thiết bị vào chuẩn (bàn phím), chuyển dịch chúng (thành số nguyên, số thực, ký tự vv..) rồi lưu trữ nó vào bộ nhớ theo các địa chỉ xác định.

Cú pháp : `scanf(« chuỗi định dạng», đối số 1, đối số 2, ...);`

Khi sử dụng hàm phải khai báo tiền xử lý `#include <stdio.h>`

- `scanf`: tên hàm, phải viết bằng chữ thường.

- chuỗi định dạng: được đặt trong cặp nháy kép (" ") là hình ảnh dạng dữ liệu nhập vào.

- đối số 1, đối số 2...: là danh sách các đối mục cách nhau bởi dấu phẩy, mỗi đối mục sẽ tiếp nhận giá trị nhập vào.

Sau đây là các dấu mô tả định dạng:

`%c` : Ký tự đơn

`%s` : Chuỗi

`%d` : Số nguyên thập phân có dấu

`%f` : Số chấm động (ký hiệu thập phân)

`%e` : Số chấm động (ký hiệu có số mũ)

`%g` : Số chấm động (`%f` hay `%g`)

`%x` : Số nguyên thập phân không dấu

`%u` : Số nguyên hex không dấu

`%o` : Số nguyên bát phân không dấu

`l`: Tiền tố dùng kèm với `%d`, `%u`, `%x`, `%o` để chỉ số nguyên dài (ví dụ : `%ld`)

Ví dụ1: `scanf("%d", &i);`

Nhập vào 12abc, biến `i` chỉ nhận giá trị 12. Nhập 3.4 chỉ nhận giá trị 3.

Ví dụ 2: `scanf("%d%d", &a, &b);`

Nhập vào 2 số `a, b` phải cách nhau bằng khoảng trắng hoặc enter.

Ví dụ 3: `scanf("%d/%d/%d", &ngay, &thang, &nam);`

Nhập vào ngày, tháng, năm theo dạng ngày/thang/nam (20/12/2002)

Lệnh xuất.

Cú pháp : `printf(“chuỗi định dạng”, đối số 1, đối số 2, ...);`

Khi sử dụng hàm phải khai báo tiền xử lý `#include <stdio.h>`

- `printf`: tên hàm, phải viết bằng chữ thường.

- đối số 1, đối số 2...: là các mục dữ kiện cần in ra màn hình. Các đối số này có thể là biến, hằng hoặc biểu thức phải được định trị trước khi in ra.

- chuỗi định dạng: được đặt trong cặp nháy kép (" "), gồm 3 loại:

- + Đối với chuỗi ký tự ghi như thế nào in ra giống như vậy.

+ Đối với những kí tự chuyển đổi dạng thức cho phép kết xuất giá trị của các đối mục ra màn hình tạm gọi là mã định dạng. (Ta đã có bảng mã định dạng ở mục 2)

* Các ký tự điều khiển và ký tự đặc biệt :

\n : Nhảy xuống dòng kế tiếp canh về cột đầu tiên.

\t : Canh cột tab ngang.

\r : Nhảy về đầu hàng, không xuống hàng.

\a : Tiếng kêu bip.

\\ : In ra dấu \

\\" : In ra dấu "

\' : In ra dấu '

%%: In ra dấu %

Ví dụ 1 : printf("\n Nang suat tang : %d % \" ",30);

→ Kết quả in ra màn hình :

"Nang suat tang : 30%"

Ví dụ 2 : float x=25.5, y=-47.335

printf("%f\n%*.2f",x,y);

→ Kết quả in ra màn hình :

25.500000

-47.34

BÀI TẬP THỰC HÀNH

1. Viết chương trình đổi một số nguyên hệ 10 sang hệ 2.
2. Viết chương trình đổi một số nguyên hệ 10 sang hệ 16.
3. Viết chương trình đọc vào 2 số nguyên và in ra kết quả của phép (+), phép trừ (-), phép nhân (*), phép chia (/). Nhận xét kết quả chia 2 số nguyên.
4. Viết chương trình nhập vào bán kính hình cầu, tính và in ra diện tích, thể tích của hình cầu đó. Hướng dẫn: $S = 4 \pi R^2$ và $V = (4/3) \pi R^3$
5. Viết chương trình nhập vào một số a bất kỳ và in ra giá trị bình phương (a^2), lập phương (a^3) của a và giá trị a^4
6. Viết chương trình đọc từ bàn phím 3 số nguyên biểu diễn ngày, tháng, năm và xuất ra màn hình dưới dạng "ngay/thang/nam" (chỉ lấy 2 số cuối của năm).
7. Viết chương trình nhập vào số giây từ 0 đến 86399, đổi số giây nhập vào thành dạng "gio:phut:giay", mỗi thành phần là một số nguyên có 2 chữ số.
Ví dụ: 02:11:05