LẬP TRÌNH C/C++

LẬP TRÌNH C/C++

01/01/2018

Khoa Công nghệ Thông tin

GIỚI THIỆU

*Giáo trình được biên soạn hoàn toàn dựa trên nhiều tài liệu của các đồng nghiệp. Giáo trình nên dùng với những cuốn sách giáo khoa hoàn chỉnh về ngôn ngữ lập trình C/C++. Giáo trình được chia thành 10 chương theo từng nội dung kiến thức, kèm theo Các đề thi mẫu. Mỗi chương gồm 2 phần: Phần lý thuyết: được tóm tắt ngắn gọn với đầy đủ ví dụ minh hoạ kèm theo. Phần bài tập: với nhiều bài tập được chia làm hai mức độ cơ bản và luyện tập nâng cao, bài tập có đánh dấu * là bài tập khó dành cho sinh viên luyện tập thêm. Phần tóm tắt: tóm tắt nội dung tóm và các thao tác mà sinh viên cần nắm hay những lưu ý của chương đó.*

Mục lục

GIỚI THIỆU ii

1 CÁC KHÁI NIỆM CƠ BẢN VỀ LẬP TRÌNH 1

1.1 KHÁI NIỆM LẬP TRÌNH 1
1.2 CÔNG NGHỆ LẬP TRÌNH TRUYỀN THỐNG 4
1.3 CÔNG NGHỆ LẬP TRÌNH HIỆN ĐẠI 4
1.3.1 Công cụ và môi trường lập trình 5
1.4 THUẬT TOÁN 5

2 LƯU ĐỒ THUẬT TOÁN 8

2.1 LÝ THUYẾT 8
2.1.1 Khái niệm 8
2.1.2 Phần mềm công cụ 8
2.1.3 Các ký hiệu 9
2.1.4 Phương pháp vẽ và thực thi 9
2.1.5 Các cấu trúc điều khiển cơ bản 9
2.2 BÀI TẬP 11
2.3 TÓM TẮT 13

3 KIỂU DỮ LIỆU CƠ BẢN VÀ CẤU TRÚC ĐIỀU KHIỂN 14

3.1 LÝ THUYẾT 14
3.1.1 Cấu trúc một chương trình C đơn giản 14
3.1.2 Các kiểu dữ liệu cơ bản trong C 15
3.1.3 Các phép toán 16

3.1.4 Các hàm thư viện cơ bản . . . 18
3.1.5 Cấu trúc tuần tự . . . 19
3.1.6 Cấu trúc rẽ nhánh . . . 20
3.1.7 Cấu trúc lặp . . . 23
3.1.8 Cấu trúc nhảy . . . 26
3.1.9 Thực thi chương trình . . . 27
3.2 BÀI TẬP . . . 28
3.3 TÓM TẮT . . . 36

4 HÀM 37
4.1 LÝ THUYẾT . . . 37
4.1.1 Khái niệm . . . 37
4.1.2 Cấu trúc một chương trình C . . . 38
4.1.3 Cách xây dựng một hàm con . . . 39
4.2 BÀI TẬP . . . 43
4.3 TÓM TẮT . . . 46

5 KIỂU DỮ LIỆU MẢNG MỘT CHIỀU 47
5.1 LÝ THUYẾT . . . 47
5.2 BÀI TẬP . . . 48
5.3 TÓM TẮT . . . 61

6 KIỂU DỮ LIỆU CHUỖI KÝ TỰ 62
6.1 LÝ THUYẾT . . . 62
6.2 BÀI TẬP . . . 65
6.3 TÓM TẮT . . . 67

7 KIỂU DỮ LIỆU MẢNG HAI CHIỀU 68
7.1 LÝ THUYẾT . . . 68
7.2 BÀI TẬP . . . 69
7.3 TÓM TẮT . . . 80

8 KIỂU DỮ LIỆU CẤU TRÚC 81
8.1 LÝ THUYẾT . . . 81
8.2 BÀI TẬP . . . 97

8.3 TÓM TẮT 102

9 KIỂU DỮ LIỆU TẬP TIN 103

9.1 LÝ THUYẾT 103

9.1.1 Khái niệm tập tin 103

9.1.2 Thao tác với tập tin 103

9.1.3 Các ví dụ minh hoạ 106

9.2 BÀI TẬP 110

9.3 TÓM TẮT 114

10 KIỂU DỮ LIỆU CON TRỎ 115

10.1 LÝ THUYẾT 115

10.2 BÀI TẬP 123

10.3 TÓM TẮT 123

11 KỸ THUẬT ĐỆ QUI 124

11.1 LÝ THUYẾT 124

11.1.1 Các loại đệ qui 125

11.1.2 Các ví dụ 130

11.2 BÀI TẬP 131

11.3 TÓM TẮT 133

12 LẬP TRÌNH ĐƠN THỂ 134

12.1 LÝ THUYẾT 134

13 MỘT SỐ ĐỀ THI MẪU 135

Tài liệu tham khảo 145

1 CÁC KHÁI NIỆM CƠ BẢN VỀ LẬP TRÌNH

An algorithm must be seen to be believed.

Donald Ervin Knuth

1.1 KHÁI NIỆM LẬP TRÌNH

Định nghĩa.

- CPU của máy tính được thiết kế để có thể thực hiện được các *chương trình mã máy* (machine code program) đã được hệ điều hành (HĐH) nạp vào RAM của máy tính.
- Chương trình mã máy thường phải tương thích với từng họ máy cụ thể, bao gồm tập hợp các chỉ thị được viết bằng các lệnh CPU của họ máy đó, được lưu trên đĩa dưới dạng một *tập tin mã thực thi* (executable program file) của HĐH cụ thể

Quy trình thực hiện

- Bước 1: *Người sử dụng* (end user) ra lệnh thực hiện chạy chương trình.
- Bước 2: HĐH nhận được lệnh sẽ thực hiện:
  - Tìm và nạp tập tin mã thực thi của chương trình (nằm trên đĩa) vào RAM của máy tính.
  - Bộ đếm lệnh của CPU (CPU program counter) được trỏ đến lệnh đầu tiên của chương trình hay còn gọi là *ngõ vào chương trình* (program entry point)
- Bước 3: CPU thực hiện từng chỉ thị một trong RAM cho đến khi gặp lệnh kết thúc:
  - Chép lệnh mã máy hiện hành vào thanh ghi lệnh.

  - Tăng bộ đếm lệnh (để trỏ đến lệnh kế tiếp).
  - Thi hành lệnh mã máy.
- Bước 4: Kết thúc thực hiện chương trình, HĐH chờ nhận lệnh mới.

Đặc điểm

- Mỗi chỉ thị của chương trình là một lệnh mã máy (một dãy các byte chỉ phù hợp với qui ước tập lệnh của một loại CPU nào đó)
- Được cấu trúc hóa theo qui ước của HĐH.
- Được chạy trên một họ CPU và HĐH cụ thể.
- Nội dung rất khó hiểu đối với người dùng máy tính, chỉ có CPU thích hợp với hiểu rõ và thi hành được.

Nhận xét

- Khó có thể sản xuất ra phần mềm bằng cách viết trực tiếp các chương trình mã máy.
- Nếu có làm được theo cách này thì
  - Giá cả sẽ rất đắt do quá khó, tốn quá nhiều thời gian và công sức.
  - Khả năng dùng lại rất giới hạn do không thể bán chongười dùng trên họ máy tính khác hay người dùng sử dụng hệ điều hành khác.

Định nghĩa.

- *Ngôn ngữ lập trình* - NNLT (programming language) là ngôn ngữ được lập trình viên sử dụng để viết chương trình cho máy tính.
- Khi một chương trình được viết bằng một NNLT nào đó thì các chỉ thị, câu lệnh trong chương trình phải tuân theo các qui tắc, các luật do NNLT đó qui định.
- Chương trình viết bằng ngôn ngữ lập trình được gọi là *chương trình nguồn* (source code program) hay *mã nguồn* (source code). Chương trình nguồn được dịch sang chương trình mã máy bằng cách chương trình dịch:
  - *Trình hợp dịch* (assembler) để dịch các chương trình hợp ngữ.
  - *Trình thông dịch* (interpreter) và *trình biên dịch* (compiler) để dịch các chương trình cấp cao.

Ví dụ. Một số ngôn ngữ lập trình thông dụng

- Ngôn ngữ cấp thấp: Hợp ngữ (assembly language)
- Ngôn ngữ cấp cao: C/C++, Java, C#, Pascal, Python, PHP, Ruby, Perl, Lisp

Định nghĩa.

- *Chương trình* (program) là một dãy các *chỉ thị* (instruction) điều khiển sự hoạt động của máy tính nhằm giải quyết một công việc nào đó.
- Người viết chương trình hay còn gọi là lập trình viên hay thảo chương viên (programmer) là những người tạo lập ra những chương trình máy tính.

Ví dụ 1.1

Một số chương trình

- Ngôn ngữ assembly

```
.model tiny
.code
org 100h
    main  proc
        mov    ah,9
        mov    dx,offset hello_message
        int    21h
        retn
        hello_message db 'Hello, world!$'
    main endp
end main
```

- Ngôn ngữ C/C++

```
#include <stdio.h>
void main(void)
{
    pritnf("Hello world!");
}
```

- Ngôn ngữ Java

```
public class Hello {
    public static void main(String argv[])
    {
        System.out.print("Hello everybody!");
```

```
    }
}
```

Đặc điểm ngôn ngữ lập trình cấp thấp

- Là NNLT phụ thuộc vào từng họ máy cụ thể, vì vậy không có tính tương thích.
- Dễ viết, đọc, sửa hơn chương trình mã máy.
- Ưu điểm là tận dụng và khai thác được tính năng của mỗi họ máy cụ thể, nhờ vậy chương trình có thể chạy nhanh hơn.

Đặc điểm ngôn ngữ lập trình cấp cao

- Được đề xuất để khắc phục các hạn chế của NNLT cấp thấp.
- Dễ dùng và dễ diễn đạt được các ý tưởng trừu tượng.
- Có tính tương thích cao (khi thay đổi dạng máy tính thì chỉ cần sửa chương trình rất ít hoặc thậm chí không cần sửa mà vẫn đảm bảo chạy đúng).

1.2 CÔNG NGHỆ LẬP TRÌNH TRUYỀN THỐNG

Đối với các NNLT cấp cao truyền thống (trước thế hệ của Java và C#), quá trình viết, dịch và chạy chương trình gồm các công đoạn như sau:

- B1. *Soạn* chương trình nguồn và lưu lên đĩa.
- B2. *Dịch* chương trình nguồn nhờ trình biên dịch.
- B3. *Nối kết* các tập tin mã trung gian tạo ra ở B2.
- B4. *Chạy* chương trình ngôn ngữ máy tạo ra ở B3.

1.3 CÔNG NGHỆ LẬP TRÌNH HIỆN ĐẠI

- Hạn chế của các chương trình cấp cao truyền thống là trình biên dịch của chúng phát sinh trực tiếp mã thực thi phụ thuộc vào mã máy tính của một họ máy tính và hệ điều hành cụ thể nên không thể mang đi sử dụng ở các hệ điều hành khác.
- NNLT hiện đại như Java hay C# trình biên dịch không dịch trực tiếp mã nguồn thành mã thực thi mà được thiết kế để có thể dịch thành *mã thực thi trừu tượng* (abstract executable code) độc lập máy và hệ điều hành.

- Do máy tính thật không thể hiểu được mã trừu tượng nên những chương trình dạng mã thực thi trừu tượng chỉ chạy được khi có sẵn máy ảo hỗ trợ cho việc thi hành loại mã thực thi đó.
- Trong các năm gần đây, các ứng dụng chạy trên web phát triển rất mạnh.
  - Chạy trên **Internet** thông qua một trình duyệt web.
  - Được viết bằng các ngôn ngữ như Python, PHP, ASP.NET, JSP, Java Script, VB Script... có tính tương thích cao, hoạt động trên bất kỳ máy tính nào có internet

1.3.1 Công cụ và môi trường lập trình

Toàn bộ qui trình biên dịch được thực một cách dễ dàng và thuận tiện nhờ vào công cụ gọi là *môi trường phát triển phần mềm* - IDE (integrated development environment)

- Soạn thảo chương trình.
- Quản lý hệ thống tập tin mã nguồn.
- Quản lý hệ thống các phiên bản của mã nguồn.
- Kiểm tra lỗi cú pháp (syntax error), biên dịch (compile), liên kết chương trình (link).
- Chạy từng dòng lệnh (debug) để tìm lỗi.

Ví dụ. Một số IDE thông dụng

- Eclipse: hỗ trợ nhiều ngôn ngữ lập trình.
- Visual Studio: hỗ trợ nghiều ngôn ngữ lập trình

1.4 THUẬT TOÁN

Các bước thiết kế chương trình

- Xác định bài toán
- Phân tích bài toán
- Thiết kế thuật toán
- Cài đặt chương trình
- Thử nghiệm chương trình

Định nghĩa. *Thuật toán* (algorithm) là tập hợp hữu hạn các *chỉ thị* được định nghĩa rõ ràng nhằm giải quyết một vấn đề cụ thể nào đó.

Tính chất. *Một thuật toán phải có các tính chất sau*

- *Tính chính xác: quá trình tính toán hay các thao tác máy tính thực hiện là chính xác.*
- *Tính rõ ràng: các câu lệnh minh bạch được sắp xếp theo thứ tự nhất định.*
- *Tính khách quan: được viết bởi nhiều người trên máy tính nhưng kết quả phải như nhau.*
- *Tính phổ dụng: có thể áp dụng cho một lớp các bài toán có đầu vào tương tự nhau.*
- *Tính kết thúc: hữu hạn các bước tính toán*

Trình bày thuật toán như thế nào?

- Trình bày bằng ngôn ngữ tự nhiên
- Trình bày bằng mã giả
- Trình bày bằng lưu đồ

Ví dụ 1.2

Hãy trình bày cách làm món sữa dâu bằng ngôn ngữ tự nhiên

Cách 1

- Lấy một ít sữa.
- Đổ nước ép dâu vào.
- Trộn hỗn hợp này và làm lạnh.

Cách 2

- Rót một ly sữa vào máy xay
- Đổ thêm vào một ít nước dâu ép
- Đóng nắm máy xay
- Mở điện và bắt đầu trộn
- Dừng máy trộn lại
- Nếu đã trộn đều thì tắt máy, ngược lại thì trộn tiếp
- Khi đã trộn xong, rót hỗn hợp vào tô và đặt vào tủ lạnh
- Để lạnh một lúc rồi lấy ra dùng

Ví dụ 1.3

Hãy trình bày cách sắp xếp một dãy số L theo thứ tự tăng dần bằng ngôn ngữ mã giả

```
MergeSort(L)
    if Size(L)>1
```

```
Split(L, L1, L2) (tách dãy L thành L1 và L2)
MergeSort(L1) (sắp xếp dãy L1)
MergeSort(L2) (sắp xếp dãy L2)
Merge(L1, L2, L) (trộn dãy L1 và L2 thành L)
```

2 LƯU ĐỒ THUẬT TOÁN

A picture is worth a thousand words

thành ngữ

Chương này sẽ trình bày các ký hiệu biểu diễn lưu đồ thuật toán, cách biểu diễn các cấu trúc điều khiển rẽ nhánh, cấu trúc lặp và các kỹ thuật liên quan đến lưu đồ thuật toán.

2.1 LÝ THUYẾT

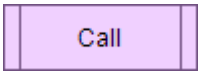
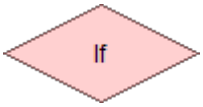
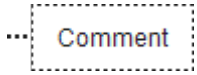
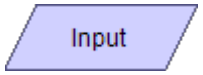
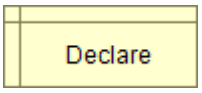
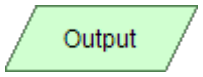
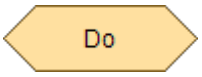
2.1.1 Khái niệm

Lưu đồ thuật toán là công cụ đồ thị dùng để biểu diễn thuật toán, việc mô tả nhập dữ liệu và xuất dữ liệu và xử lý dữ liệu thông qua các ký hiệu hình học.

2.1.2 Phần mềm công cụ

Có rất nhiều công cụ để vẽ lưu đồ từ offline cho đến online. Giáo trình sử dụng phần mềm Flowgorithm để vẽ và thực thi lưu đồ

2.1.3 Các ký hiệu

Lệnh	Ký hiệu	Lệnh	Ký hiệu
Assignment		For	
Call		If	
Comment		Input	
Declare		Output	
Do		While	

2.1.4 Phương pháp vẽ và thực thi

- Vẽ từ trên xuống
- Chạy bắt đầu từ Begin và kết thúc tại End
- Đi theo hướng mũi tên

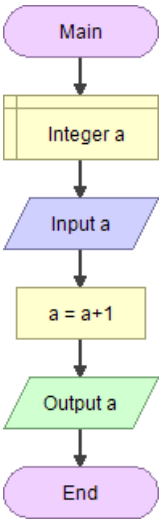
2.1.5 Các cấu trúc điều khiển cơ bản

Cấu trúc tuần tự

Tuần tự thực thi tiến trình. Mỗi lệnh được thực thi theo một chuỗi từ trên xuống, xong lệnh này rồi chuyển xuống lệnh kế tiếp.

Ví dụ 2.1

Nhập vào một số nguyên a và xuất ra màn hình với giá trị tăng lên 1

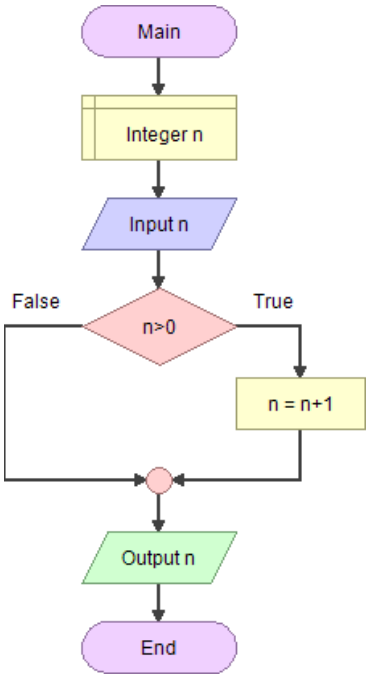


Cấu trúc rẽ nhánh

Điểm quyết định cho phép chọn một trong hai trường hợp đúng hay sai

Ví dụ 2.2

Nhập vào số nguyên n. Kiểm tra nếu $n > 0$ tăng n lên 1 đơn vị

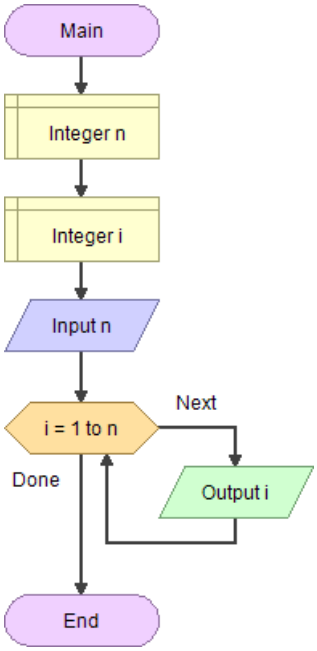


Cấu trúc lặp

Thực hiện liên tục 1 lệnh hay tập lệnh với số lần lặp dựa vào điều kiện. Lặp sẽ kết thúc khi điều kiện được thỏa.

Ví dụ 2.3

Nhập vào số nguyên n. Xuất ra màn hình từ 1 đến n.



Các ví dụ thực hành

2.1. Giải và biện luận phương trình: $ax + b = 0$.

2.2. Tính tổng: $S(n) = 1 + 2 + 3 + \ldots + n$ với $n > 0$

2.3. Tính tổng: $S(n) = \frac{1}{2} + \frac{2}{3} + \frac{3}{4} + \ldots + \frac{2n+1}{2n+2}$ với $n > 0$

2.4. Tính tổng: $S(n) = 1 - 2 + 3 - \ldots + (-1)^{n+1}n$ với $n > 0$

2.2 BÀI TẬP

Bài tập cơ bản

2.5. Nhập vào hai số x, y. Xuất ra màn hình tổng, hiệu, tích, thương của hai số trên.

2.6. Nhập vào số nguyên n, kiểm tra xem n chẵn hay lẻ và xuất ra màn hình.

2.7. Nhập vào ba cạnh a, b, c của tam giác. Xuất ra màn hình tam giác đó thuộc loại tam giác gì? (thường, cân, vuông, đều hay vuông cân).

2.8. Nhập vào số nguyên n. Xuất n ra màn hình (nếu n chẵn thì gấp đôi giá trị).

2.9. Nhập vào số nguyên n. Nếu $n > 5$ thì tăng n lên 2 đơn vị và trả về giá trị n, ngược lại trả về giá trị 0.

2.10. Tính $n!$, với $n \geq 0$

2.11. Tính $P(n) = 1.3.5...(2n+1)$, với $n \geq 0$

2.12. Tính $S(n) = 1 + 3 + 5 + ... + (2n+1)$, với $n \geq 0$

2.13. Tính $S(n) = 1 - 2 + 3 - 4 + ... + (-1)^{n+1}n$, với $n > 0$

2.14. Tính $S(n) = 1 + 1.2 + 1.2.3 + ... + 1.2.3...n$, với $n > 0$

2.15. Tính $S(n) = 1^2 + 2^2 + 3^2 + ... + n^2$, với $n > 0$

2.16. Tính

$$S(n) = 1 + \frac{1}{2} + \frac{1}{3} + ... + \frac{1}{n}$$

với $n > 0$

2.17. (*) Tính

$$S(n) = 1 + \frac{1}{1+2} + \frac{1}{1+2+3} + ... + \frac{1}{1+2+3+...+n}$$

với $n > 0$

2.18. Tính $P(x, y) = x^y$

2.19. Tính $S(n) = 1 + (1+2) + (1+2+3) + ... + (1+2+3+...+n)$ với $n > 0$

2.20. Cho số nguyên n. Tính trị tuyệt đối của n.

2.21. Cho số nguyên dương n gồm k chữ số. Tìm chữ số có giá trị lớn nhất.

2.22. Đếm số lượng ước số chẵn của số nguyên dương n.

2.23. In ra chữ số đầu tiên của số nguyên dương n gồm k chữ số.

2.24. Cho 2 số nguyên dương a, b. Tìm USCLN của a và b.

2.25. Cho 2 số nguyên dương a, b. Tìm BSCNN của a và b.

2.26. Cho số nguyên dương x. Kiểm tra xem x có phải là số nguyên tố không?

2.27. Cho số nguyên dương x. Kiểm tra x có phải là số chính phương không?

2.28. Cho số nguyên dương x. Kiểm tra xem x có phải là số hoàn thiện không?

Luyện tập và nâng cao

2.29. Tính $S(n) = 1 + 2^2 + 3^3 + ... + n^n$, với $n > 0$

2.30. Tính

$$S(n) = \frac{1}{2} + \frac{2}{3} + ... + \frac{n}{n+1}$$

với $n > 0$

2.31. Tính

$$S(n) = 1 + \frac{1}{2!} + \frac{1}{3!} + ... + \frac{1}{n!}$$

với $n > 0$

2.32. Tính

$$S(n) = 1 + \frac{1+2}{2!} + \frac{1+2+3}{3!} + \ldots + \frac{1+2+3+...+n}{n!}$$

với $n > 0$

2.33. Giải và biện luận phương trình bậc hai $ax^2 + bx + c = 0$

2.34. Giải và biện luận phương trình trùng phương bậc bốn $ax^4 + bx^2 + c = 0$

2.35. (*) Tính $S(n) = \sqrt{n + \sqrt{n-1 + \sqrt{n-2 + ... + \sqrt{1}}}}$ với $n > 0$

2.36. (**) Tính $S(n) = \sqrt{1 + \sqrt{2 + \sqrt{3 + ... + \sqrt{n}}}}$ với $n > 0$

2.3 TÓM TẮT

Lưu đồ thuật toán rất là một công cụ hữu ích trong việc mô tả cách giải quyết của một bài toán. Việc mô tả này rất trực quan thông qua các ký hiệu hình học, đây là giai đoạn đầu tiên trước khi bắt tay vào lập trình trên một ngôn ngữ lập trình cụ thể. Khi xây dựng lưu đồ thuật toán, chúng ta cần chú ý một vài điểm sau: Một lưu đồ phải có điểm bắt đầu và điểm kết thúc. Phải có dữ liệu vào, dữ liệu ra sau khi xử lý tính toán. Tại mỗi vị trí quyết định lựa chọn rẽ nhánh phải ghi rõ điều kiện đúng hoặc sai thì đi theo nhánh nào.

3 KIỂU DỮ LIỆU CƠ BẢN VÀ CẤU TRÚC ĐIỀU KHIỂN

> C is quirky, flawed, and an enormous success.
>
> Dennis M. Ritchie

Chương này trình bày các kiểu dữ liệu cơ bản và các phép toán tương ứng trong ngôn ngữ C, các cấu trúc rẽ nhánh, lặp. Mô tả cách hoạt động và hướng dẫn chạy từng bước chương trình.

3.1 LÝ THUYẾT

3.1.1 Cấu trúc một chương trình C đơn giản

- Khai báo thư viện
- Khai báo biến toàn cục
- Chương trình chính (`main`)

Ví dụ 3.1

Một chương trình đơn giản

```
// Khai bao thu vien
#include <stdio.h>
#include <conio.h>
// Khai bao bien toan cuc
int a, b;
// Chuong trinh chinh
```