# CHƯƠNG 3: CẤU TRÚC RẼ NHÁNH CÓ ĐIỀU KIỆN

Ngôn ngữ lập trình C cung cấp các loại cấu trúc rẽ nhánh sau.

|  |  |  |
| --- | --- | --- |
| STT | Phân loại | Chi tiết |
| 1 | Câu lệnh if | Lệnh if |
| Lệnh if … else |
| Lệnh if … else if … else |
| Các lệnh if lồng nhau |
| 2 | Câu lệnh switch | Lệnh switch case |
| Lệnh switch case default |
| Cách lệnh switch lồng nhau |
| 3 | Câu lệnh if và switch kết hợp |  |

## CÂU LỆNH IF

### Câu lệnh if

#### Khái niệm

Câu lệnh if được sử dụng để kiểm tra một biểu thức lý luận nào đó có đúng hay không. Nếu đúng thì thực thi khối lệnh; ngược lại nếu sai thì bỏ qua khối lệnh đó.

Trong lập trình C, bất kỳ giá trị nào = 0 và  "/0" (NULL) là sai, còn lại là giá trị đúng.

### Phân loại

Có 3 dạng câu lệnh if trong C

|  |  |  |
| --- | --- | --- |
| **STT** | **Phân loại** | **Chi tiết** |
| 1 | Câu lệnh if | Lệnh if (if thiếu) |
| Lệnh if ... else |
| Lệnh if ... elseif … else |
| Các lệnh if lồng nhau. |

Sử dụng if để chỉ định một khối mã sẽ được thực thi, nếu một điều kiện cụ thể là true

Sử dụng else để chỉ định một khối mã sẽ được thực thi, nếu điều kiện tương tự là false

Sử dụng else if để chỉ định một điều kiện mới để kiểm tra, nếu điều kiện đầu tiên là false

#### Lệnh if

|  |
| --- |
| if (Biểu thức luận lý ){      // Khối lệnh 1  } |

|  |
| --- |
| **NẾU** biến số thỏa điều kiện ...  **THÌ** thực hiện ... |

Trong đó:

* if phải viết bằng chữ cái thường
* Biểu thức luận lý phải là đúng hoặc sai

**Lưu ý:**

* Không đặt dấu chấm phẩy sau câu lệnh if.
* Biểu thức luận lý phải đặt trong cặp dấu ( ).
* Nếu khối lệnh từ 2 lệnh trở lên thì phải đặt trong dấu { }.

**Ví dụ**

|  |
| --- |
| if (tuoi >= 18){      printf ("Bạn là người trưởng thành");  } |

Hoặc bạn có thể làm như sau kết quả sẽ hoàn toàn giống nhau nhưng chúng ta ưu tiên cách 1 hơn vì nó sẽ dễ hơn khi bạn thực hiện nhiều dòng code

|  |
| --- |
| if (tuoi >= 18)      printf ("Bạn là người trưởng thành"); |

**Lưu đồ**

Diagram

Description automatically generated

#### Lệnh if … else

|  |
| --- |
| if (Biểu thức luận lý 1){      // Câu lệnh 1  }  else {      // Câu lệnh 2  } |

|  |
| --- |
| **NẾU** biến số thỏa điều kiện...  **THÌ** thực hiện ...  **NẾU KHÔNG** hãy thực hiện ... |

Trong đó:

* Từ khóa if, else phải viết bằng chữ thường.
* Kết quả của biểu thức luận lý phải là đúng hoặc sai.

**Lưu ý:**

* Không đặt dấu chấm phẩy sau câu lệnh if, else.
* Biểu thức luận lý phải đặt trong cặp dấu ( ).
* Nếu khối lệnh từ 2 lệnh trở lên thì phải đặt trong dấu { }.

**Lưu đồ**

Diagram

Description automatically generated

Nếu biểu thức logic đúng thì thực hiện khối lệnh 1 và thoát khỏi if. Ngược lại thực hiện khối lệnh 2 và thoát khỏi if.

Ví dụ:

|  |
| --- |
| if (tuoi >= 18){      printf ("Ban la nguoi truong thanh");  }  else {      printf ("Ban la con nit");  } |

Ta có thể hiểu đơn giản: nếu như biến số tuổi có giá trị lớn hơn hoặc bằng 18, sẽ hiển thị « Ban la nguoi truong thanh »

Nếu không hiển thị « Ban con la con nit »

# 2.1.3. Lệnh if … else if … else

|  |
| --- |
| if (Biểu thức luận lý 1){      // Khối lệnh 1  }  else if (Biểu thức luận lý 2){      // Khối lệnh 2  }  else{      // Khối lệnh 3  } |

|  |
| --- |
| **NẾU** biến số thỏa điều kiện 1  THÌ thực hiện việc 1  **NẾU KHÔNG NẾU** biến số thỏa điều kiện 2  **THÌ** thực hiện việc 2  **NẾU KHÔNG** thực hiện việc 3 |

Trong đó:

* Từ khóa if, else if, else phải viết bằng chữ thường
* Kết quả của biểu thức luận lý phải là đúng hoặc sai.

**Lưu ý:**

* Không đặt dấu chấm phẩy sau câu lệnh if, else if, else.
* Biểu thức luận lý phải đặt trong cặp dấu ( )
* Nếu khối lệnh từ 2 lệnh trở lên thì phải đặt trong dấu { }

**Lưu đồ:**

Diagram

Description automatically generated

Nếu biểu thức logic 1 đúng thì thực hiện khối lệnh 1 và thoát khỏi cấu trúc if.

Ngược lại Nếu biểu thức logic 2 đúng thì thực hiện khối lệnh 2 và thoát khỏi cấu trúc if …

Ngược lại Nếu biểu thức logic n-1 đúng thì thực hiện khối lệnh n-1 và thoát khỏi cấu trúc if

Ngược lại thì thực hiện khối lệnh n.

Ví dụ

|  |
| --- |
| if (tuoi >= 18){      printf ("Ban la nguoi truong thanh");  }  else if (tuoi > 4) {      printf ("Ban la con nit");  }  else {      printf ("Ban la embe");  } |

#### 2. Các lệnh if lồng nhau

Trong lập trình C, ta có thể sử dụng các lệnh if lồng nhau. Thường cấu trúc if lồng nhau càng nhiều cấp thì độ phức tạp càng cao, chương trình chạy càng chậm và trong lúc lập trình dễ bị nhầm lẫn.

**Lưu ý:**

Các lệnh if … else lồng nhau thì else sẽ luôn luôn kết hợp với if nào chưa có else gần nhất. Vì vậy khi gặp những lệnh if không có else, Bạn phải đặt chúng trong những khối lệnh rõ ràng để tránh bị hiểu sai câu lệnh.

# BÀI 2: LỆNH SWITCH

#### 1. Câu lệnh switch

##### 1.1 Khái niệm

Lệnh switch case là một cấu trúc điều khiển và rẽ nhánh hoàn toàn có thể được thay thế bằng cấu trúc if else. Tuy nhiên, nó cũng có mặt hạn chế là kết quả của biểu thức phải là giá trị hằng nguyên (có giá trị cụ thể).

Việc sử dụng switch sẽ giúp code của chúng ta dễ viết, dễ đọc và có hiệu năng tốt hơn so với sử dụng if else trong trường hợp có nhiều điều kiện có thể xảy ra.

##### 1.2 Phân loại

Câu lệnh switch gồm 2 loại

* Lệnh switch case hay còn gọi là switch thiếu.
* Lệnh switch case default hay còn gọi là switch đủ.

|  |  |
| --- | --- |
| **Tên** | **Cú pháp** |
| switch thiếu | switch (Biểu thức) {      case <Giá trị 1> :          <Khối lệnh 1>          break;      case <Giá trị 2> :          <Khối lệnh 2>          break;      case <Giá trị n> :          <Khối lệnh 3>          break;  } |
| switch đủ | switch (Biểu thức) {      case <Giá trị 1> :          <Khối lệnh 1>          break;      case <Giá trị 2> :          <Khối lệnh 2>          break;      case <Giá trị n> :          <Khối lệnh 3>          break;      default :          <Khối lệnh n>          break;  } |

Trong đó:

* Từ khóa switch, case, default, break phải viết bằng chữ thường.
* Biểu thức bắt buộc là giá trị hằng, có thể là biểu thức nhưng kết quả cần là hằng số nguyên (char, int, long,…)
* Khối lệnh 1, 2,… n có thể gồm nhiều câu lệnh, nhưng không cần đặt trong cặp dấu { }

**Lưu đồ:**

Diagram, schematic

Description automatically generated

Khi chương trình thực thi, **Biểu thức** sẽ được so sánh lần lượt với giá trị của các **case.**

* Nếu có 1 case nào đó khớp giá trị thì khối lệnh tương ứng của case đó sẽ được thực hiện cho tới khi gặp lệnh break. Do đó, nếu chúng ta không sử dụng break thì tất cả các case kể từ case khớp giá trị đều được thực hiện.
* Khối lệnh sau default sẽ được thực hiện nếu không có case nào khớp giá trị với **Biểu thức**.

#### 2. Các lệnh switch lồng nhau

Các lệnh switch lồng nhau là lệnh switch chứa ít nhất 1 lệnh switch khác bên trong nó. Các lệnh switch lồng nhau càng nhiều cấp độ phức tạp càng cao, chương trình chạy càng chậm và trong lúc lập trình dễ bị nhầm lẫn

Bài tập 1: Viết chương trình nhập vào tháng và in ra quý. Sử dụng switch thiếu.

Nhập vào giá trị tháng, kiểm tra xem tháng có hợp lệ (trong khoảng 1 đến 12). Nếu hợp lệ in ra quý tương ứng (13: quý 1, 46: quý 2, 79: quý 3, 1012: quý 4).

|  |
| --- |
| #include<stdio.h>  int main(){      int thang;      printf ("Nhap thang: ");      scanf ("%d", &thang);      if (thang >= 1 && thang <= 12){          switch (thang){          case 1:          case 2:          case 3:              printf("Quy 1");              break;          case 4:          case 5:          case 6:              printf("Quy 2");              break;          case 7:          case 8:          case 9:              printf("Quy 3");              break;          case 10:          case 11:          case 12:              printf("Quy 4");              break;      }  }  else {      printf ("Xin moi nhap lai thang");  }  } |