

**HỌC VIỆN KTQS**  
**KHOA CÔNG NGHỆ THÔNG TIN**

***Chương 6.* Các cấu trúc điều khiển**

Học phần: LẬP TRÌNH CƠ BẢN

# Tài liệu tham khảo

---

- ▶ *Kỹ thuật lập trình C: cơ sở và nâng cao*, Phạm Văn Ất (thư viện). Chương 5
- ▶ *The C programming language 2nd Edition*, Brian Kernighan and Dennis Ritchie, Prentice Hall Software Series. Chương 3.

# Nội dung

---

- Tìm hiểu về cấu trúc lựa chọn
  - Lệnh if
    - Lệnh if – else
    - Lệnh nhiều if
    - Lệnh if lồng nhau
  - Lệnh switch

# Nội dung

---

- ▶ Tìm hiểu về vòng lặp ‘for’ trong C
- ▶ Làm việc với toán tử dấu phẩy (,)
- ▶ Tìm hiểu về các vòng lặp lồng nhau
- ▶ Tìm hiểu về vòng lặp ‘while’ và ‘do-while’
- ▶ Làm việc với các lệnh break và continue
- ▶ Tìm hiểu về hàm exit()

# Câu lệnh lựa chọn

---

- ▶ Các câu lệnh lựa chọn cho phép chúng ta thay đổi hướng thực hiện của chương trình
- ▶ Một câu lệnh lựa chọn trả về giá trị đúng hoặc sai
- ▶ Ví dụ: Để xác định một số là số chẵn hay số lẻ chúng ta tiến hành như sau:
  - 1) Nhập vào một số
  - 2) Chia số đó cho 2 để xác định số dư
  - 3) Nếu số dư là 0, số đó là “SỐ CHẴN”
  - 4) Ngược lại số dư không bằng 0, số đó là “SỐ LẺ”

# Các cấu trúc lựa chọn

---

C cung cấp hai dạng câu lệnh lựa chọn

**Lệnh if**

**Lệnh switch**

# Lệnh if

---

- ▶ Cú pháp:

```
if (expression)  
    statement;
```

- ▶ Nếu biểu thức của lệnh if có giá trị đúng (true), khối lệnh theo sau lệnh if được thực thi

# Lệnh if (tt.)

---

Chương trình hiển thị các giá trị dựa vào một điều kiện.

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

```
void main() {
```

```
    int x, y;
```

```
    char a = 'y' ;
```

```
    x = y = 0;
```

```
    if (a == 'y') {
```

```
        x += 5;
```

```
        printf("The numbers are %d and \t%d", x, y) ;
```

```
    }
```

```
}
```

**Ví dụ**



# Lệnh if – else

---

```
if (expression)
    statement1;
else
    statement2;
```

## Lệnh if – else (tt.)

---

- Nếu biểu thức của if trả về giá trị **true**, khối lệnh theo sau lệnh if được thực thi
- Nếu biểu thức của if **không** trả về giá trị true thì các lệnh theo sau else được thực thi
- Lệnh else là một tùy chọn. Lệnh else được dùng để thực thi các lệnh khi biểu thức trong if trả về giá trị **false**

# Lệnh if – else (tt.)

---

Chương trình xác định một số là số chẵn hay lẻ

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

```
main() {
```

```
    int num ,res ;
```

```
    printf("Enter a number :");
```

```
    scanf ("%d" , &num) ;
```

```
    res = num % 2;
```

```
    if (res == 0)
```

```
        printf("Then number is Even");
```

```
    else
```

```
        printf("The number is Odd");
```

```
}
```

**Ví dụ**

# Lệnh if-else-if

---

## Cú pháp:

```
if (expression1)
    statement1;
else if (expression2)
    statement2;
else if (expression3)
    statement3;
...
else statementn;
```

# Lệnh if–else–if (tt.)

---

- Lệnh **if - else - if** còn được gọi là lệnh **if-else-if bậc thang**
- Các biểu thức được xác định giá trị theo hướng từ trên xuống

# Lệnh if–else–if (tt.)

---

## Chương trình hiển thị thông báo dựa vào một giá trị :

```
#include <stdio.h>
#include <conio.h>
main() {
    int x;
    x = 0;
    clrscr ();
    printf("Enter Choice (1 - 3) : ");
    scanf("%d", &x);
    if (x == 1)
        printf ("\nChoice is 1");
    else if ( x == 2)
        printf ("\nChoice is 2");
    else if ( x == 3)
        printf ("\nChoice is 3");
    else printf ("\nInvalid Choice ");
}
```

**Ví dụ**

# Lệnh if lồng nhau

---

- ▶ Lệnh **if** lồng nhau là một lệnh **if** được đặt trong một lệnh **if** hoặc **else** khác
- ▶ Trong C, một lệnh **else** luôn kết hợp với một lệnh **if** gần nhất cùng khối lệnh nếu lệnh **else** đó chưa được kết hợp với một lệnh **if** nào khác

# Lệnh if lồng nhau (tt.)

---

- Cú pháp:

```
if (exp1)
{
    if (exp2) statement1;
    if (exp3) statement2;
    else statement3;           /*with if (exp3) */
}
else statement4;              /* with if (exp1) */
```

- Lệnh else bên trong được kết hợp với **if(exp3)**
- Theo chuẩn ANSI, một trình biên dịch hỗ trợ ít nhất là 15 mức lồng nhau



# Lệnh if lồng nhau (tt.)

---

```
#include <stdio.h>
#include <conio.h>
main () {
    int x, y;
    x = y = 0;
    clrscr ();
    printf ("Enter Choice (1 - 3) : ");
    scanf ("%d", &x);
    if (x == 1) {
        printf("\nEnter value for y (1 - 5) : ");
        scanf ("%d", &y);
        if (y <= 5)
            printf("\nThe value for y is : %d", y);
        else
            printf("\nThe value of y exceeds 5 ");
    }
    else    printf ("\nChoice entered was not 1");
}
```

**Ví dụ**

# Ví dụ

---

- ▶ - Ví dụ 1: Tìm min, max của 3 số a,b,c
- ▶ - Ví dụ 2: chương trình nhập số nguyên dương, kiểm tra số đó có là số chính phương hay không;
- ▶ - Ví dụ 3: chương trình tìm nghiệm của phương trình bậc 2

# Lệnh switch

---

- Lệnh **switch** là một bộ lựa chọn đa hướng, nó so sánh giá trị của một biểu thức với một danh sách các hằng số nguyên hoặc hằng ký tự
- Khi gặp một sự so sánh khớp, các lệnh kết hợp với hằng đó được thực thi

# Lệnh switch (tt.)

---

- ▶ Cú pháp:
  - ▶ **witch** (*expression*) {
    - case** *constant1*:  
*block of instructions 1*  
**break**;
    - case** *constant2*:  
*block of instructions 2*  
**break**;
    - 
    - 
    - 
    - default**:  
*default block of instructions*
  - }

# Lệnh switch (tt.)

---

Chương trình kiểm tra một ký tự thường được nhập vào là một nguyên âm, là ký tự 'z' hay là một phụ âm

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

main () {
    char ch;
    clrscr ();
    printf ("\nEnter a lower cased
    alphabet (a - z) : ");
    scanf ("%c", &ch);
```

**Ví dụ**

còn tiếp ...

# Lệnh switch (tt.)

---

```
if (ch < 'a' || ch > 'z')
    printf("\nCharacter not a lower cased alphabet");
else
    switch (ch) {
        case 'a' :
        case 'e' :
        case 'i' :
        case 'o' :
        case 'u' :
            printf("\nCharacter is a vowel");
            break;
        case 'z' :
            printf ("\nLast Alphabet (z) was entered");
            break;
        default :
            printf("\nCharacter is a consonant");
            break;
    }
}
```

## Ví dụ

---

- ▶ Ví dụ (minh hoạ cấu trúc switch): nhập biểu thức đơn giản dạng  $a \otimes b$ . tính và đưa ra kết quả
- ▶ ( $\otimes$ : có thể là phép toán số học, hoặc quy ước x, : )

# Vòng lặp là gì?

---

**Một đoạn mã lệnh trong chương trình thực hiện lặp đi lặp lại cho đến khi một điều kiện xác định được thỏa mãn**



## 3 kiểu cấu trúc vòng lặp

---

**Vòng lặp for**

**Vòng lặp while**

**Vòng lặp do....while**

# Vòng lặp for

---

## Cú pháp:

```
for (initialize counter; conditional test; re-evaluation parameter){  
    statement  
}
```

- *initialize counter* là một lệnh gán để khởi tạo biến điều khiển của vòng lặp trước khi đi vào vòng lặp
- *conditional test* là một biểu thức quan hệ để chỉ định khi nào vòng lặp sẽ kết thúc
- ▶ *re-evaluation parameter* định nghĩa cách thức thay đổi của biến điều khiển vòng lặp mỗi khi vòng lặp được thực thi

## Vòng lặp for (tt.)

---

- Ba phần của vòng lặp **for** phải được phân cách bởi dấu chấm phẩy(;)
  - ▶ Phần lệnh tạo nên thân vòng lặp có thể là một lệnh đơn hoặc một lệnh ghép (một tập nhiều lệnh)
- Vòng lặp **for** tiếp tục được thực thi khi biểu thức kiểm tra điều kiện vẫn có giá trị **true**. Khi điều kiện trở thành **false**, chương trình thực hiện lệnh theo sau vòng lặp **for**

# Vòng lặp for - Ví dụ

---

```
/*This program demonstrates  
the for loop in a C program*/  
#include <stdio.h>  
main()    {  
    int count;  
    printf("\tThis is a \n");  
    for(count = 1;count <=6;count++)  
        printf("\n\t\t nice");  
    printf("\n\t\t world. \n");  
}
```

# Toán tử dấu phẩy

Vòng lặp **for** có thể được mở rộng bằng cách chứa nhiều giá trị khởi tạo và nhiều biểu thức tăng/tri trong đặc tả của vòng lặp for

**Cú pháp:** `exprn1 , exprn2 ;`

```
#include <stdio.h>
main()
{
    int i, j , max;
    printf("Please enter the maximum value \n");
    printf("for which a table can be printed:");
    scanf("%d", &max);
    for(i = 0 , j = max ; i <=max ; i++, j--)
        printf("\n%d + %d = %d", i, j, i + j);
}
```

# Vòng lặp for lồng nhau

---

**Các vòng lặp for lồng nhau** khi nó có dạng như sau

```
for (i = 1; i < max1; i++) {  
    ...  
    for (j = 0; j <= max2; j++) {  
        ...  
    }  
    ...  
}
```

# Vòng lặp for lồng nhau -

## Ví dụ

---

```
#include <stdio.h>
main() {
    int i, j, k;
    i = 0;
    printf("Enter no. of rows :");
    scanf("%d", &i);
    printf("\n");
    for (j = 0; j < i ; j++) {
        printf("\n");
        for (k = 0; k <= j; k++) /*inner for loop*/
            printf("*");
    }
}
```

## Ví dụ

---

- ▶ - Ví dụ 1: In ra 10 số tự nhiên đầu tiên
- ▶ - Ví dụ 2: In các kí tự từ A - Z
- ▶ - Ví dụ 3: Giải bài toán vui trăm trâu trăm cỏ
- ▶ - Ví dụ 4: Nhập số  $n$ , in các số hoàn chỉnh  $\leq n$



# Vòng lặp while

---

## Cú pháp

```
while (condition)  
statement ;
```

**Vòng lặp while lặp lại các lệnh trong khi một biểu thức điều kiện mang giá trị True**

# Vòng lặp while - Ví dụ

---

```
/*A simple program using the while loop*/
#include <stdio.h>
main() {
    int count = 1;
    while( count <= 10)    {
        printf("\n This is iteration
                                   %d\n", count) ;
        count++;
    }
    printf("\n The loop is completed.\n");
}
```

## Ví dụ

---

- ▶ - Ví dụ 1 (minh hoạ cấu trúc while): tính  $n!$ ;
- Ví dụ 2: tính tổng , với  $i = 1 \rightarrow n$ ;
- Ví dụ 3: kiểm tra số nguyên tố
- ▶ - Ví dụ 4: Tìm USCLN của 2 số  $a, b$

# Vòng lặp do...while

Cú pháp

do{

statement;

} while (condition);

- Trong vòng lặp **do while** phần thân của vòng lặp được thực thi trước khi biểu thức điều kiện được kiểm tra
- Khi điều kiện mang giá trị False, vòng lặp **do while** sẽ được kết thúc, và điều khiển chuyển đến lệnh xuất hiện ngay sau lệnh **while**

# Vòng lặp do...while - Ví dụ

---

```
#include <stdio.h>
main () {
    int num1, num2;
    num2 = 0;
    do {
        printf( "\nEnter a number : " );
        scanf( "%d", &num1 );
        printf( " No. is %d", num1 );
        num2++;
    } while (num1 != 0);
    printf ( "\nThe total numbers entered were
            %d", --num2 );
    /*num2 is decremented before printing because
    count for last integer (0) is not to be considered
    */
}
```

## Ví dụ

---

- ▶ - Ví dụ 1 (minh hoạ cấu trúc do while): Viết chương trình nhập dãy các số nguyên dương từ bàn phím, cho tới khi nhập số âm thì kết thúc nhập, in giá trị lớn nhất;

# Các lệnh chuyển điều khiển

---

## `return` **expression**

- Lệnh `return` được sử dụng để trở về từ một hàm
- Thực hiện lệnh `return` để trở về vị trí mà tại đó hàm được gọi
- Lệnh `return` có thể có một giá trị đi cùng, giá trị này được trả về cho chương trình gọi

# Các lệnh chuyển điều khiển (tt.)

## goto label

- Lệnh goto chuyển điều khiển đến một câu lệnh bất kỳ khác bên trong cùng một hàm trong một chương trình C
- Điều này thật ra vi phạm đến qui luật của một ngôn ngữ lập trình cấu trúc.
- Chúng làm giảm độ tin cậy của chương trình và chương trình khó bảo trì.



# Các lệnh chuyển điều khiển (tt.)

## break

- Lệnh `break` được sử dụng để kết thúc một mệnh đề *case* trong câu lệnh *switch*
- Nó cũng có thể được sử dụng để kết thúc ngang giữa vòng lặp
- Khi gặp lệnh `break`, vòng lặp sẽ kết thúc ngay và điều khiển được chuyển đến lệnh kế tiếp bên ngoài vòng lặp

# Lệnh break – Ví dụ

---

```
#include <stdio.h>
main () {
    int count1, count2;
    for(count1 = 1, count2 = 0;
        count1 <=100; count1++) {
        printf("Enter %d count2: ", count1);
        scanf("%d", &count2);
        if(count2==100) break;
    }
}
```

# Các lệnh chuyển điều khiển (tt.)

## *continue*

- Lệnh *continue* dùng để bắt đầu thực hiện lần lặp kế tiếp của vòng lặp
- Khi gặp lệnh *continue*, các câu lệnh còn lại trong thân vòng lặp bị bỏ qua và điều khiển được chuyển đến lần lặp kế tiếp

# Lệnh continue – Ví dụ

---

```
#include <stdio.h>
main ()      {
    int num;
    for(num = 1; num<=100; num++) {
        if(num % 9 == 0)
            continue;
        printf("%d\t", num) ;
    }
}
```

# Các lệnh chuyển điều khiển (tt)

## **hàm** `exit()`

- Hàm `exit()` được sử dụng để thoát khỏi chương trình
- Sử dụng hàm này sẽ kết thúc ngay chương trình và điều khiển được chuyển về cho hệ điều hành

# Tóm tắt nội dung

---

- ▶ Cấu trúc tuần tự
- ▶ Cấu trúc rẽ nhánh (if, switch)
- ▶ Cấu trúc lặp
  - ▶ for
  - ▶ while
  - ▶ do while

# Thảo luận

---

- ▶ So sánh các vòng lặp
- ▶ Ý nghĩa sử dụng của từng loại vòng lặp
- ▶ Hạn chế khi sử dụng break

# CÂU HỎI VÀ BÀI TẬP

---

- ▶ Bài 1 - Nhập dãy có  $n$  số thực, hãy xác định trật tự của dãy đó (tăng, giảm, hay không có thứ tự)
- ▶ Bài 2 -Viết chương trình nhập số  $n$ , tìm và in các số nguyên tố thuộc  $2..n$
- ▶ Bài 3: Kiểm tra số fibonaxi
- ▶ Bài 4: Kiểm tra số hoàn chỉnh
- ▶ Bài 5: Tính  $\sin(x)$  theo công thức khai triển Taylor với độ chính xác cho trước
- ▶ Bài 6: Nhập dãy có  $n$  số thực, hãy xác định trật tự của dãy đó (tăng, giảm, điều hay không có thứ tự)
- ▶ Bài 7: Nhập  $n$ , in  $n$  số fibonaxi đầu tiên
- ▶ Bài 8: Nhập số  $n$ , in các số nguyên tố  $2..n$
- ▶ Bài 9: Viết chương trình nhập dãy các số nguyên dương từ bàn phím, cho tới khi nhập số âm thì kết thúc nhập, tìm giá trị lớn nhất.



# CÂU HỎI VÀ BÀI TẬP

---

- ▶ Bài 10: tính  $\cos(x)$  theo công thức khai triển Taylor
- ▶ Bài 11: Tính tổng  $1/(i!)$  với  $i: 1 \div n$
- ▶ Bài 12: Tìm  $n$  số nguyên tố đầu tiên
- ▶ Bài 13: Nhập  $n$  số thực, tìm và in giá trị lớn nhất, nhỏ nhất
- ▶ Bài 14: Viết chương trình nhập dãy các số nguyên dương từ bàn phím, cho tới khi nhập số âm thì kết thúc nhập, tìm giá trị lớn nhất
- ▶ Bài 15: Viết chương trình chọn ngẫu nhiên một kí tự  $c$  nào đó, cho phép lặp nhập một kí tự từ bàn phím, nếu trùng với  $c$  thì thông báo chọn đúng và kết thúc
- ▶ Bài 16: Viết chương in bảng cửu chương ra màn hình
- ▶ Bài 17: Tính đại lượng với  $n$  cho trước  $S_{\sqrt{2 + \sqrt{4 + \dots + \sqrt{2(n-1) + \sqrt{2n}}}}}$

---

# HỎI VÀ ĐÁP