CƠ SỞ LẬP TRÌNH

CÁC PHẦN TỬ CƠ BẢN CỦA NGÔN NGỮ C

Nội dung

- Các thành phần cơ bản
- Cấu trúc chương trình C
- Các kiểu dữ liệu cơ sở
- Câu lệnh - biểu thức
- Thứ tự ưu tiên các phép toán
- Vào - ra dữ liệu trong C

1. Các thành phần cơ bản

- Bộ từ vựng của C
  - Các chữ cái hoa: A, B, C, …, Z
  - Các chữ cái thường: a, b, c, …, z
  - Các chữ số : 0, 1, 2, 4, 5, 6, 7, 8, 9
  - Các ký hiệu toán học : + – * / = < > ( )
  - Các ký tự đặc biệt : . , : ; [ ] % \ # $ ‘ ^ & @
  - Ký tự gạch nối _ và khoảng trắng ‘ ’, dấu tab, xuống dòng

1. Các thành phần cơ bản (tt)

- Từ khóa (keyword)
  - Các từ dành riêng trong ngôn ngữ, mỗi từ có tác dụng và ý nghĩa cụ thể
  - Không thể sử dụng từ khóa để đặt tên cho biến, hàm, tên chương trình con.
  - Một số từ khóa thông dụng:
    - const, enum, signed, struct, typedef, unsigned…
    - char, double, float, int, long, short, void
    - case, default, else, if, switch
    - do, for, while
    - break, continue, goto, return

1. Các thành phần cơ bản (tt)

- Tên/Định danh (Identificater)
  - Tên là dãy kí tự liền nhau gồm các chữ cái a..z, A..Z, các chữ số 0..9, và dấu gạch nối.
  - Mọi tên đều phải khai báo trước khi sử dụng
  - Tên trong C phân biệt chữ HOA, thường
  - Độ dài tối đa mặc định là 32 kí tự
- Quy tắc đặt tên
  - Tên không được trùng với các từ khoá
  - Không được bắt đầu bằng chữ số
  - Không chứa kí tự đặc biệt như dấu cách, dấu chấm
  - Tên phải gợi nhớ về đối tượng được đặt tên
  - Cùng phạm vi không được đặt 2 tên trùng nhau

1. Các thành phần cơ bản (tt)

- Ví dụ Tên/Định danh (Identifier)
  - Các tên hợp lệ: GiaiPhuongTrinh, Bai_Tap1, PI
  - Các tên không hợp lệ:
    - 1A bắt đầu bằng chữ số
    - PI$ chứa kí hiệu $
    - Giai phuong trinh chứa dấu cách
    - char trùng từ khoá char
  - Phân biệt chữ hoa chữ thường, do đó các tên sau đây khác nhau:
    - A, a
    - BaiTap, baitap, BAITAP, bAItaP, …
- Thường dùng chữ HOA đặt tên cho hằng, chữ thường cho các đối tượng khác.

1. Các thành phần cơ bản (tt)

- Dấu chấm phẩy ;
  - Dùng để phân cách các câu lệnh.
  - Ví dụ: printf(“Hello World!”); printf(“\n”);
- Câu chú thích
  - Đặt giữa cặp dấu /* */ hoặc // (C++)
  - Ví dụ: /*Ho & Ten: NVA*/, // MSSV: 0712078
- Hằng ký tự và hằng chuỗi
  - Hằng ký tự: ‘A’, ‘a’, …
  - Hằng chuỗi: “Hello World!”, “Nguyen Van A”
  - Chú ý: ‘A’ khác “A”

2. Cấu trúc chung chương trình C

```
#include <…>      /*Gọi các tệp tiền xử lý */
#define /* Định nghĩa */
typedef /*Định nghĩa kiểu */
int x;            /* Khai báo biến ngoài */
const …           /*Khai báo hằng */
/*Khai báo các hàm, có thể có hoặc không */
Kiểu_dữ_liệu tên_hàm(các tham số);
{     Khai báo các biến, hằng
      Các lệnh của hàm
      return(); /*Trả lại giá trị */
} ...
main()      /* Bắt buộc phải có hàm main */
{     Khai báo các biến, hằng
      Các lệnh của hàm
      return (); /*Có thể có hoặc không */
}
```

Ví dụ chương trình C

- Ví dụ 1: Viết ra màn hình dòng chữ

  CHAO MUNG DEN VOI NGON NGU C

```
#include <stdio.h>
#include <conio.h>
#include <stdlib.h>
int main()
{
        system("cls"); /*Xoá màn hình */
        printf("CHAO MUNG DEN VOI NGON NGU C");
        getch(); /*Dừng màn hình */
        return 0;
}
```

Ví dụ chương trình C

- Ví dụ 2: Tính chu vi và diện tích hình tròn với bán kính r nhập từ bàn phím.

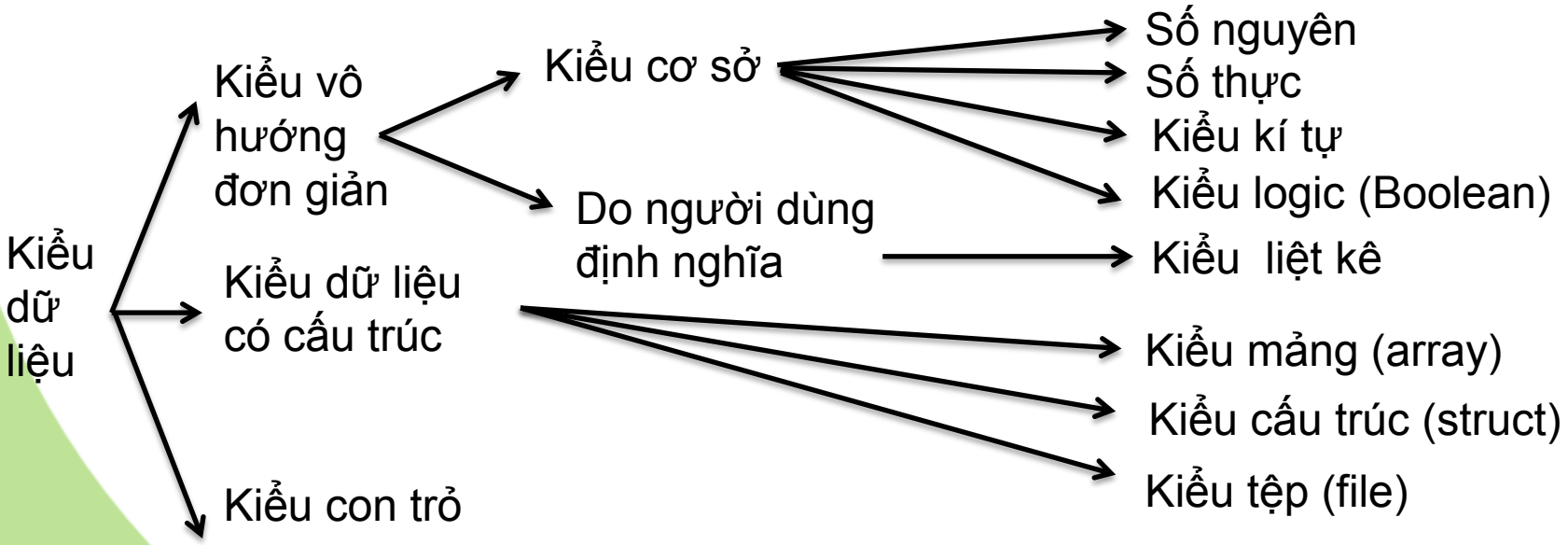
```
#include <stdio.h> /*Thư viện vào ra chuẩn */
#include <conio.h>
#include <stdlib.h>
#include <math.h> /*Thư viện hàm toán học*/
int main()
{       float r,cv,dt; /*Khai báo biến*/
        system("cls"); /*Xoá màn hình */
        printf("Nhap ban kinh: "); scanf("%f",&r);
        cv=2*M_PI*r; dt=M_PI*r*r; /*Tính chu vi, diện tích*/
        printf("Chu vi: %0.2f",cv); printf("Dien tich: %0.2f",dt);
        getch(); /*Dừng màn hình */
        return 0;
}
```

Một số quy tắc khi viết chương trình

- Mỗi câu lệnh có thể viết trên một hay nhiều dòng, nhưng phải kết thúc bằng dấu ;
- Để báo cho C biết một chuỗi kí tự vẫn còn ở dòng dưới, thêm dấu \ trước khi xuống dòng
  - Ví dụ:
```
printf("CHAO MUNG \
        DEN VOI NGON NGU C");
```
- Lời chú thích có thể viết trên 1 hoặc nhiều dòng, đặt giữa cặp dấu /*...*/
- Các lệnh theo cùng nhóm phải thẳng hàng theo chiều dọc

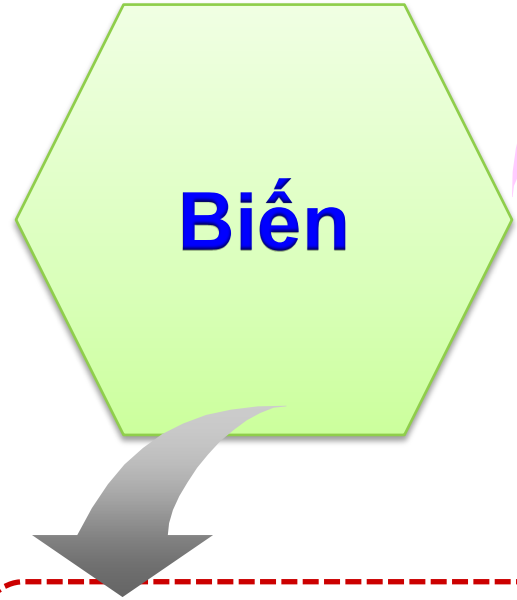
3. Các kiểu dữ liệu cơ sở

- Kiểu dữ liệu (data type) là:
  - Một tập hợp các giá trị mà một biến thuộc kiểu đó có thể nhận được,
  - Trên đó xác định một số phép toán



Biến

- Là đại lượng có thể thay đổi được giá trị



Ví dụ:

```
int i;
int j, k;
unsigned char dem;
float ketqua, delta;
```

Cú pháp

<kiểu dữ liệu> <danh sách các biến>;

- Trong C, giá trị `i` được chứa trong ô nhớ có địa chỉ `&i`

Hằng

- Là đại lượng có giá trị không đổi

Hằng thường

Ví dụ

```
const int A = 1506;
const int B = 01506;
const int C = 0x1506;
const float D = 15.06e-3;
const char RC='\r'
```

Hằng tượng trưng

Ví dụ

```
#define MAX 100
#define PI 3.14
#define TRUE 1
#define FALSE 0
```

Các kiểu dữ liệu cơ sở (tự đọc)

- C có 4 kiểu cơ sở
  - Kiểu số nguyên: giá trị của nó là các số nguyên như 2912, -1706, …
  - Kiểu số thực: giá trị của nó là các số thực như 3.1415, 29.12, -17.06, …
  - Kiểu ký tự: 256 ký tự trong bảng mã ASCII.
  - Kiểu boolean: giá trị đúng hoặc sai.

Kiểu số nguyên

- Các kiểu số nguyên (có dấu)
  - n bit có dấu: $-2^{n-1} \ldots +2^{n-1} - 1$

Kiểu (Type)	Độ lớn (Byte)	Miền giá trị (Range)
char	1	–128 … +127
int	2	–32.768 … +32.767
short	2	–32.768 … +32.767
long	4	–2.147.483.648 … +2.147.483.647

Kiểu số nguyên (tt)

- Các kiểu số nguyên (không dấu)
  - n bit không dấu: 0 … $2^n - 1$

Kiểu (Type)	Độ lớn (Byte)	Miền giá trị (Range)
unsigned char	1	0 … 255
unsigned int	2	0 … 65.535
unsigned short	2	0 … 65.535
unsigned long	4	0 … 4.294.967.295

Kiểu số nguyên (tt)

- Các phép tính số học với số nguyên

Phép toán	Kí hiệu	Ví dụ	Ví dụ bằng số
Cộng	+	x+y	
Trừ	-	x-y	
Nhân	*	x*y	
Chia lấy phần nguyên	/	x/y	3/2=1 chứ không phải là 1.5
Chia lấy số dư	%	x%y	5%3 = 2

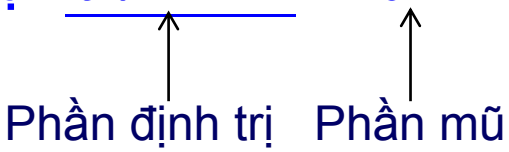
- Chú ý:
  - Chia 2 số nguyên là số nguyên, muốn là số thực phải viết `(float)x/y`
  - Thận trọng tránh hiện tượng tràn số

Kiểu số nguyên (tt)

- Biểu diễn số nguyên dạng hệ đếm 16 (Hexa)
  - Bắt đầu bằng kí tự `0x` hoặc `0X`
  - Ví dụ: `65` được viết là `0x41` hoặc `0X41`
    `15` được viết là `0xF` hoặc `0XF`
- Biểu diễn số nguyên dạng hệ đếm 8 (Octa)
  - Bắt đầu bằng kí tự `0`
  - Ví dụ: `65` được viết là `0101`
    `15` được viết là `017`
- Hằng số nguyên định trước kiểu
  - Thêm một kí tự cuối vào số: **L** (long), **U** (unsigned integer, **UL** (unsigned long)
  - Ví dụ: `50000U, 012345L, 0x50000U`

Kiểu số thực

- Dạng viết bình thường
  - Ví dụ: `3.14  3.0  -24.12345  -0.453`
- Dạng viết khoa học
  - Gồm: phần định trị và phần mũ viết sau chữ **E** (hoặc **e**), giữa chúng không có khoảng cách
  - Ví dụ: `6.2144E+02`



  - Chú ý:
    - Nếu không có phần mũ thì phần định trị bắt buộc phải có dấu **.**
    - Có thể không cần số 0 ở đầu (vd: **.1212**)
    - **KHÔNG** tồn tại phép % cho số thực

Kiểu số thực

Các kiểu số thực

Kiểu (Type)	Độ lớn (Byte)	Miền giá trị (Range)
float	4	1.24E-38 … 3.4E38 Độ chính xác khoảng 7 chữ số
double	8	2.2E-308 … 1.8E308 Độ chính xác khoảng 15 chữ số
long double	10	3.4E4932…3.4E4932 Độ chính xác khoảng 19 chữ số

Hằng số thực định trước kiểu

- Thêm kí tự cuối: F (float), L (long double)
- Ví dụ: `0.1234567-20L` `4E12F`

Kiểu ký tự

- Đặc điểm
  - Tên kiểu: char
  - Miền giá trị: 256 ký tự trong bảng mã ASCII.
  - Chính là kiểu số nguyên do:
    - Không lưu trực tiếp ký tự mà chỉ lưu mã ASCII của ký tự đó.
- Ví dụ
  - Lưu số 65 tương đương với ký tự ‘A’
  - Lưu số 97 tương đương với ký tự ‘a’
- Hằng kí tự
  - Đặt giữa hai dấu phẩy trên
  - Ví dụ: `'a'  'A'  'z'`

Kiểu ký tự (tt)

- Biểu diễn một kí tự trong bảng mã ASCII
  - \xHHH (HHH là giá trị số Hexa của kí tự)
  - \DDD (DDD là giá trị số Octa của kí tự)
  - Ví dụ: 'A' được viết dưới dạng `\x41` hoặc `\101`

Kí tự	Dãy mã	Giá trị trong bảng ASCII
Xoá trái	\b	x08
Nhảy cách ngang	\t	x09
Xuống dòng	\n	x0A
Dấu "	\"	x22
Dấu '	\'	x27
Dấu \	\\	x5C
Mã null	\0	x00

Kiểu ký tự (tt)

- Các hàm xử lí kí tự
  - `toASCII(c)`: chuyển c thành giá trị mã ASCII
  - `tolower(c)`: chuyển thành chữ thường
  - `toupper(c)`: chuyển thành chữ hoa
- Hằng xâu kí tự
  - Hằng xâu kí tự được viết trong cặp nháy kép “”
  - Xâu kí tự được lưu trữ trong một mảng ô nhớ liền nhau và có ô cuối cùng chứa mã số 0 (null)
    - Ví dụ: Xâu “`Viet nam`” được lưu là:

V	i	e	t		n	a	m	\0

  - Hằng xâu kí tự không được viết trong biểu thức số học

Kiểu Boolean

- Đặc điểm
  - C ngầm định một cách không tường minh:
    - false (sai): giá trị 0.
    - true (đúng): giá trị khác 0, thường là 1.
  - C++: bool
- Ví dụ
  - 0 (false), 1 (true), 2 (true), 2.5 (true)
  - 1 > 2 (0, false), 1 < 2 (1, true)

4. Câu lệnh – Biểu thức

- Biểu thức
  - Tạo thành từ các toán tử (Operator) và các toán hạng (Operand).
  - Toán tử tác động lên các giá trị của toán hạng và cho giá trị có kiểu nhất định.
  - Toán tử: +, –, *, /, %….
  - Toán hạng: hằng, biến, lời gọi hàm...
- Ví dụ
  - 2 + 3, a / 5, (a + b) * 5, …

Câu lệnh

- Khái niệm
  - Là một chỉ thị trực tiếp, hoàn chỉnh nhằm ra lệnh cho máy tính thực hiện một số tác vụ nhất định nào đó.
  - Các câu lệnh cách nhau bằng dấu chấm phẩy ;
  - Trình biên dịch bỏ qua các khoảng trắng (hay tab hoặc xuống dòng) chen giữa lệnh.
- Ví dụ

```
a=2912;
a = 2912;
a
=
2912;
```

Câu lệnh (tt)

- Phân loại
  - Câu lệnh đơn: chỉ gồm một câu lệnh.
  - Câu lệnh phức (khối lệnh): gồm nhiều câu lệnh đơn được bao bởi { và }
- Ví dụ

```
a = 2912;                    // Câu lệnh đơn

{                            // Câu lệnh phức/khối lệnh
      a = 2912;
      b = 1706;
}
```

Phép gán

- Phép gán giá trị đơn giản

  <Tên biến> = <Biểu thức>

  Ví dụ: `i=3;`

  `i=i+4;`

  - Giá trị của biểu thức bên phải dấu gán = được đặt vào ô nhớ của biến nằm bên trái dấu gán.
  - Vế trái chỉ có thể là tên của một biến hoặc là giá trị dạng một địa chỉ ô nhớ.

- Phép gán kép

  `a=b=c=3;` Gán giá trị 3 cho cả 3 biến a,b,c

  `a=b+(c=3);` Gán 3 cho c, sau đó cộng với b và gán kết quả nhận được cho a

Sự hiệu chỉnh dữ liệu khi tính toán

- Máy tự đông chuyển kiểu đơn giản lên kiểu cao hơn để quy đổi kiểu kết quả, theo thứ tự:

  int → long → float → double → long double

- Chuyển đổi cho kiểu kí tự char
  - Chuyển đổi qua lại giữa char và int
  - Ví dụ: 'A' + 1 = 66
- Cố ý chuyển đổi kiểu giá trị (typecast)
  - Cú pháp: `kiểu(biến)` hoặc `(kiểu)biến`
  - Ví dụ: f = float(1) / 2; g = float(1 / 2);

5. Thứ tự ưu tiên các phép toán

Toán tử toán học

- Toán tử 1 ngôi
  - Chỉ có một toán hạng trong biểu thức.
  - ++ (tăng 1 đơn vị), -- (giảm 1 đơn vị)
  - Đặt trước toán hạng
    - Ví dụ ++x hay --x: thực hiện tăng/giảm trước.
  - Đặt sau toán hạng
    - Ví dụ x++ hay x--: thực hiện tăng/giảm sau.
- Ví dụ
  - x = 10; y = x++; // y = 10 và x = 11
  - x = 10; y = ++x; // x = 11 và y = 11

Các toán tử toán học

- Toán tử 2 ngôi
  - Có hai toán hạng trong biểu thức.
  - +, –, *, /, % (chia lấy phần dư)
- Ví dụ
  - a = 1 + 2; b = 1 – 2; c = 1 * 2; d = 1 / 2;
  - e = 1*1.0 / 2;
  - h = 1 % 2;
  - x = x * (2 + 3*5); ⇔ x *= 2 + 3*5;

Phép gán mở rộng

- C cho phép viết lệnh gán mở rộng theo quy định:

```
x+=y  ⇔  x= x+y
x-=y  ⇔  x= x-y
x*=y  ⇔  x= x*y
x/=y  ⇔  x= x/y
x%=y  ⇔  x= x%y
x>>=y ⇔  x= x>>y
x<<=y ⇔  x= x<<y
x&=y  ⇔  x= x&y
x|=y  ⇔  x= x|y
x^=y  ⇔  x= x^y
```

Các toán tử trên bit

Các toán tử trên bit

- Tác động lên các bit của toán hạng (nguyên).
- & (and), | (or), ^ (xor), ~ (not hay lấy số bù 1)
- >> (shift right), << (shift left)
- Toán tử gộp: &=, |=, ^=, ~=, >>=, <<=

&	0	1
0	0	0
1	0	**1**

|	0	1
0	0	**1**
1	**1**	**1**

^	0	1
0	0	**1**
1	**1**	0

~	0	1
	1	0

>>	**a>>n = a/2^n**
<<	**a<<n = a*2^n**

Các toán tử trên bit

- Ví dụ

```
void main()
{
        int a = 5;  // 0000 0000 0000 0101
        int b = 6;  // 0000 0000 0000 0110

        int z1, z2, z3, z4, z5, z6;
        z1 = a & b; // 0000 0000 0000 0100
        z2 = a | b; // 0000 0000 0000 0111
        z3 = a ^ b; // 0000 0000 0000 0011
        z4 = ~a;    // 1111 1111 1111 1010
        z5 = a >> 2;// 0000 0000 0000 0001
        z6 = a << 2;// 0000 0000 0001 0100
}
```

Các toán tử quan hệ

- Các toán tử quan hệ
  - So sánh 2 biểu thức với nhau
  - Cho ra kết quả 0 (hay false nếu sai) hoặc 1 (hay true nếu đúng)
  - ==, >, <, >=, <, <=, !=
- Ví dụ
  - s1 = (1 == 2);s2 = (1 != 2);
  - s3 = (1 > 2); s4 = (1 >= 2);
  - s5 = (1 < 2); s6 = (1 <= 2);

Các toán tử logic

Các toán tử logic

- Tổ hợp nhiều biểu thức quan hệ với nhau.
- && (and), || (or), ! (not)

&&	0	1
0	0	0
1	0	**1**

||	0	1
0	0	**1**
1	**1**	**1**

- Ví dụ
  - s1 = (1 > 2) && (3 > 4);
  - s2 = (1 > 2) || (3 > 4);
  - s3 = !(1 > 2);

Toán tử điều kiện

- Toán tử điều kiện
  - Đây là toán tử 3 ngôi (gồm có 3 toán hạng)
  - <biểu thức 1> ? <biểu thức 2> : <biểu thức 3>
    - <biểu thức 1> đúng thì giá trị là <biểu thức 2>.
    - <biểu thức 1> sai thì giá trị là <biểu thức 3>.
- Ví dụ
  - s1 = (1 > 2) ? 2912 : 1706;
  - int s2 = 0;
  - 1 < 2 ? s2 = 2912 : s2 = 1706;

Toán tử phẩy

- Toán tử phẩy
  - Các biểu thức đặt cách nhau bằng dấu ,
  - Các biểu thức con lần lượt được tính từ trái sang phải.
  - Biểu thức mới nhận được là giá trị của biểu thức bên phải cùng.
- Ví dụ
  - x = (a++, b = b + 2);
  - ⇔ a++; b = b + 2; x = b;

Độ ưu tiên của các toán tử

Toán tử	Độ ưu tiên
() [] ->	→
! ++ -- - ~ sizeof() (toán tử 1 ngôi)	←
* / %	→
+ -	→
<< >>	→
< <= > >=	→
== !=	→
&	→
|	→
^	→
&&	→
||	→
?:	←
= += -= *= /= %= &= ^= |= <<= >>=	←

Độ ưu tiên của các toán tử

- Quy tắc thực hiện
  - Thực hiện biểu thức trong ( ) sâu nhất trước.
  - Thực hiện theo thứ tự ưu tiên các toán tử.

  => Tự chủ động thêm ( )
- Ví dụ
  - n = 2 + 3 * 5;

    => n = 2 + (3 * 5);
  - a > 1 && b < 2

    => (a > 1) && (b < 2)

Viết biểu thức cho các mệnh đề

- x lớn hơn hay bằng 3

  x >= 3

- a và b cùng dấu

  ((a>0) && (b>0)) || ((a<0) && (b<0))

  (a>0 && b>0) || (a<0 && b<0)

- p bằng q bằng r

  (p == q) && (q == r) hoặc (p == q && q == r)

- –5 < x < 5

  (x > –5) && (x < 5) hoặc (x > –5 && x < 5)

6. Vào – Ra dữ liệu trong C

Xuất dữ liệu

- Thư viện
  - #include <stdio.h> (standard input/output)
- Cú pháp
  - printf(“dãy mã quy cách”, dãy các biểu thức)
  - dãy mã quy cách là dãy các định dạng được đặt trong cặp nháy kép “ ”.
    - Văn bản thường (literal text)
    - Ký tự điều khiển (escape sequence)
    - Đặc tả (conversion specifier)

Mã quy cách

- Văn bản thường (literal text)
  - Được xuất y hệt như lúc gõ trong chuỗi định dạng.
- Ví dụ
  - Xuất chuỗi Hello World
    - ➔ printf(“Hello ”); printf(“World”);
    - ➔ printf(“Hello World”);
  - Xuất chuỗi a + b
    - ➔ printf(“a + b”);

Mã quy cách (tt)

- Ký tự điều khiển (escape sequence)
  - Gồm dấu \ và một ký tự như trong bảng sau:

Ký tự điều khiển	Ý nghĩa
\a	Tiếng chuông
\b	Lùi lại một bước
\n	Xuống dòng
\t	Dấu tab
\\	In dấu \
\?	In dấu ?
\”	In dấu “

- Ví dụ
  - printf(“\t”); printf(“\n”);
  - printf(“\t\n”);

Mã quy cách (tt)

- Đặc tả (conversion specifier)
  - Gồm dấu % và một ký tự.
  - Xác định kiểu của biến/giá trị muốn xuất.
  - Các đối số chính là các biến/giá trị muốn xuất, được liệt kê theo thứ tự cách nhau dấu phẩy.

Đặc tả	Ý nghĩa	
%c	Ký tự	char
%d, %ld	Số nguyên có dấu	char, int, short, long
%f, %lf	Số thực	float, double
%s	Chuỗi ký tự	char[], char*
%u %lu	Số nguyên không dấu	unsigned int/short/long
%x, %X	Số nguyên dạng Hexa	
%o	Số nguyên dạng Octa	
%e, %E	Số thực dạng mũ	

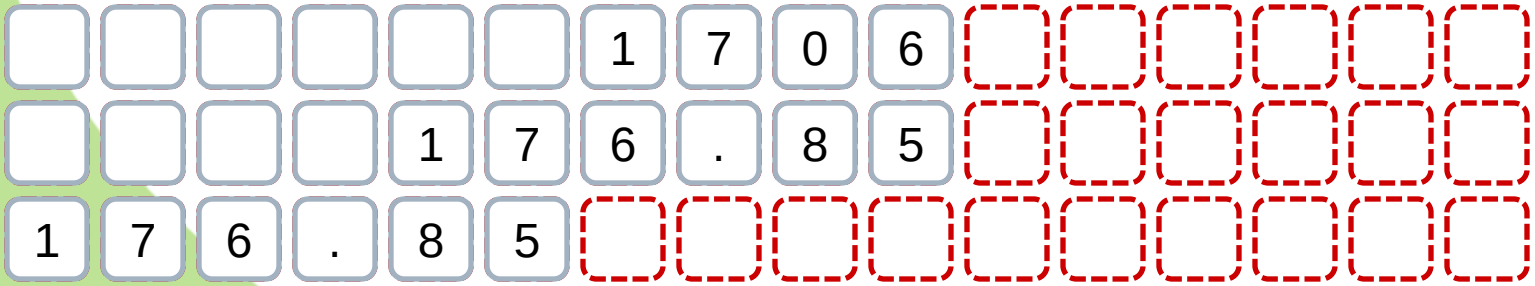
Mã quy cách (tt)

- Ví dụ
  - int a = 10, b = 20;
  - printf("%d", a); ➔ Xuất ra 10
  - printf("%d", b); ➔ Xuất ra 20
  - printf("%d %d", a, b); ➔ Xuất ra 10 20
  - float x = 15.06;
  - printf("%f", x); ➔ Xuất ra 15.060000
  - printf("%f", 1.0/3); ➔ Xuất ra 0.333333

Định dạng xuất

- Cú pháp
  - Định dạng xuất số nguyên: %nd
  - Định dạng xuất số thực: %n.kf

```
int a = 1706;
float x = 176.85;
printf("%10d", a);printf("\n");
printf("%10.2f", x);printf("\n");
printf("%.2f", x);printf("\n");
```



Mã quy cách (tt)

- Phối hợp các thành phần
  - int a = 1, b = 2;
  - Xuất 1 cong 2 bang 3 và xuống dòng.
    - printf("%d", a); // Xuất giá trị của biến a
    - printf(" cong "); // Xuất chuỗi " cong "
    - printf("%d", b); // Xuất giá trị của biến b
    - printf(" bang "); // Xuất chuỗi " bang "
    - printf("%d", a + b); // Xuất giá trị của a + b
    - printf("\n"); // Xuất điều khiển xuống dòng \n

    ➔ printf("%d cong %d bang %d\n", a, b, a+b);

Nhập dữ liệu

- Thư viện
  - #include <stdio.h> (standard input/output)
- Hàm scanf()
  - scanf("dãy mã quy cách", dãy các địa chỉ các biến);
  - Danh sách các biến: là tên các biến sẽ chứa giá trị nhập và có dấu & đặt trước

Mã quy cách	Ý nghĩa
%c	Ký tự char
%d	Số nguyên int
%u	Số nguyên unsigned int
%hd, %hu	Số nguyên short int/ unsigned int
%ld %lu	Số nguyên long int, unsiged long
%f, %e	Số thực
%lf hoặc %lu	Số thực double
%s	Xâu không chứa dấu cách, dùng với địa chỉ xâu

Nhập dữ liệu (tt)

- Nguyên tắc đọc
  - Số: máy nhảy qua các kí tự là các dấu chấm câu cho đến khi gặp kí tự là chữ số, đọc đến khi gặp kí tự không là chữ số.
  - Xâu kí tự: đọc đúng số kí tự mà ta yêu cầu
- Vai trò của dấu cách trong mã định dạng
  - Gặp dấu cách trong mã định dạng, máy nhảy qua các dấu cách để đọc số/kí tự
- Khuôn đọc
  - Đọc một số trong phạm vi m chữ số gõ vào
  - Ví dụ: `scanf("%3d%3d",&n,&p)`
- Xoá bộ nhớ đệm: `fflush(stdin);`

Nhập dữ liệu (tt)

- Hàm getchar()
  - Đọc một kí tự từ bàn phím
- Hàm getch()
  - Đọc kí tự từ bàn phím ngay khi gõ vào không đợi ấn phím Enter và không hiển thị ra màn hình.
  - Đọc thẳng từ bàn phím, không qua bộ nhớ đệm
  - Dùng để dừng chương trình xem kết quả
- Hàm getche()
  - Giống getch(), nhưng hiển thị kí tự lên màn hình
- Hàm gets()
  - Đọc một xâu kí tự cho đến khi gõ Enter

Bài tập lý thuyết

1. Trình bày các kiểu dữ liệu cơ sở trong C và cho ví dụ.
2. Trình bày khái niệm về biến và cách sử dụng lệnh gán.
3. Phân biệt hằng thường và hằng ký hiệu. Cho ví dụ minh họa.
4. Trình bày khái niệm về biểu thức.

   Tại sao nên sử dụng cặp ngoặc đơn.
5. Trình bày cách định dạng xuất.

Bài tập thực hành

6. Nhập năm sinh của một người và tính tuổi của người đó.
7. Nhập 2 số a và b. Tính tổng, hiệu, tính và thương của hai số đó.
8. Nhập tên sản phẩm, số lượng và đơn giá. Tính tiền và thuế giá trị gia tăng phải trả, biết:
   a. tiền = số lượng * đơn giá
   b. thuế giá trị gia tăng = 10% tiền

Bài tập thực hành

9. Nhập điểm thi và hệ số 3 môn Toán, Lý, Hóa của một sinh viên. Tính điểm trung bình của sinh viên đó.
10. Nhập bán kính của đường tròn. Tính chu vi và diện tích của hình tròn đó.
11. Nhập vào số xe (gồm 4 chữ số) của bạn. Cho biết số xe của bạn được mấy nước?