Ngôn ngữ lập trình C
Mảng và các phép toán

MẢNG – Kiểu dữ liệu có cấu trúc

- Mảng : một tập hợp nhiều phần tử có cùng tên và cùng kiểu giá trị, các giá trị trong mảng gọi là phần tử của mảng.
- Để xác định một mảng:
  - Kiểu mảng (int, float, char…)
  - Tên mảng
  - Số chiều và kích thước mỗi chiều
- Mảng thường được dùng để lưu dãy số, ma trận

Mảng một chiều

- Khai báo mảng một chiều:

  Kiểu tên_mảng[n]

  n: số phần tử của mảng
- Để xác định một phần tử trong mảng nào ta chỉ cần xác định rõ tên mảng và vị trí của nó trong mảng.

  tên_mảng[vị trí]

- Vị trí các phần tử trong mảng bắt đầu từ 0.

Khai báo mảng một chiều

Ví dụ 1:

```
char    ar[10];
```

mảng tên ar sẽ quản lý 10 biến kiểu char có tên và thứ tự vị trí như sau:

ar[0]	ar[1]	ar[2]	ar[3]	ar[4]	ar[5]	ar[6]	ar[7]	ar[8]	ar[9]

Để truy nhập phần tử thứ i trong mảng ar: ta viết ar[i]

Ví dụ: ar[3] = 5;

Ví dụ 2:

```
int    ar[6];
```

mảng tên ar sẽ quản lý 6 biến kiểu int có tên và thứ tự vị trí như sau:

ar[0]	ar[1]	ar[2]	ar[3]	ar[4]	ar[5]

Mảng một chiều - Khai báo và khởi gán giá trị

Ví dụ 3:

char str[10] = “Ha Noi”;

khai báo và khởi gán giá trị cho mảng tên str kiểu char chứa tối đa 10 kí tự, các phần tử trong mảng str có giá trị như sau:

str[0]	str[1]	str[2]	str[3]	str[4]	str[5]	str[6]	str[7]	str[8]	str[9]
‘H’	‘a’	‘ ’	‘N’	‘o’	‘i’	‘\0’			

‘\0’ là dấu hiệu kết thúc chuỗi kí tự, các kí tự (nếu có) sau kí tự này không có giá trị trong chuỗi str.

Khai báo và khởi gán giá trị

Ví dụ 4: int ar[5] = {6, 2, 8, 7, 4};
khai báo và khởi gán giá trị mảng tên ar có 5 phần tử:
ar[0] = 6; ar[1] = 2; ar[2] = 8; ar[3] = 7; ar[4] = 4

Ví dụ 5: int ar[8] = {6, 2, 8, 7, 4};
khai báo và khởi gán giá trị mảng tên ar có 8 phần tử:
ar[0] = 6; ar[1] = 2; ar[2] = 8; ar[3] = 7; ar[4] = 4;
Các biến ar[5], ar[6], ar[7] chưa được gán giá trị.

Ví dụ 6: int ar[3] = {6, 2, 8, 7, 4};
khai báo và khởi gán giá trị mảng tên ar có 3 phần tử:
ar[0] = 6; ar[1] = 2; ar[2] = 8;
hai phần tử dưới đây có giá trị nhưng không chịu sự quản lý của mảng ar
ar[3] = 7; ar[4] = 4
Khi thực hiện các câu lệnh đối với ar[3], ar[4] kết quả nhận được của chương trình có thể không đoán trước được (ngoài tầm kiểm soát)

Mảng hai chiều

- Khai báo:

  Kiểu tên_mảng[m][n];

  m,n : số phần tử của mỗi chiều

- Ví dụ:

  int x[5][5];

  float a[4][2];

Các phần tử của mảng a:

a[0][0]	a[0][1]
a[1][0]	a[1][1]
a[2][0]	a[2][1]
a[3][0]	a[3][1]

Mảng hai chiều: Khởi gán giá trị

- Ví dụ :

```
float a[3][4] = { {3  4  5  1}
                  {2.5  4  3  6}
                  {10  0.5  6.7}
                  }
```

Một số chú ý

- Trong bộ nhớ, các phần tử của mảng hai chiều được sắp xếp theo hàng.
- Chỉ số của mảng phải có kiểu int không vượt quá kích thước chiều tương ứng. Số chỉ số phải bằng số chiều
  - Ví dụ đúng: a[i][j]; x[1][3]
  - Ví dụ sai: a[2]; x[0][1][2]
- Biểu thức dùng làm chỉ số có thể thực
  - Ví dụ: b[2.4] = b[2]
- Khi chỉ số mảng vượt quá kích thước mảng, máy không báo lỗi và sẽ truy cập vùng nhớ ngoài mảng gây rối chương trình

6. Các phép toán số học

Kí hiệu	Ý nghĩa
( ), [ ]	ngoặc nhóm biểu thức, vị trí của biến (phần tử) trong mảng
+, –, *, /, %	Cộng , trừ, nhân, chia, Chia modul (lấy số dư): 13%5 = 3; 13%4 = 1
++, – –	phép toán tăng 1 đơn vị, giảm 1 đơn vị
=, +=, – =	phép gán giá trị trực tiếp, cộng thêm, trừ bớt
*=, /=, %=	phép gán giá trị nhân với, chia cho, chia lấy modul cho
?:	Biểu thức có điều kiện (thường sử dụng với phép gán)

Ví dụ: a += 3; tương đương với a = a + 3;
a = 2 > 3? 4 : 5; kết quả a = 5.
a = 2 < 3? 4 : 5; kết quả a = 4.

6. Các phép toán số học

- Có thể viết ++, – – trước hoặc sau tên biến
  - a ++ và ++ a là như nhau
  - a – – và – – a là như nhau
- Khi ++, – – có mặt trong biểu thức thì vị trí của nó sẽ ảnh hưởng đến thứ tự thực hiện phép toán trong biểu thức.
- Ví dụ 1:

  a = 5;
  b = 5 + (a++); Thứ tự: b = 5 + a ➔ a++
- Ví dụ 2:

  a = 5;
  b = 5 + (++a); Thứ tự: ++a ➔ b = 5 + a

7. Các phép toán quan hệ so sánh

Kí hiệu	Ý nghĩa
==	so sánh bằng
!=	so sánh không bằng (khác)
<, <=	so sánh nhỏ hơn, nhỏ hơn hoặc bằng
>, >=	so sánh lớn hơn, lớn hơn hoặc bằng

- Kết quả phép toán quan hệ so sánh sẽ là
  - giá trị 1 nếu đúng
  - giá trị 0 nếu sai.

Ví dụ:

(4 > 5) cho kết quả là 0

((4 > 5) == 0) cho kết quả là 1

8. Các phép toán quan hệ logic

gồm có phép toán:
AND (&&), OR (||), NOT (!)

Kết quả phép toán logic sẽ là
giá trị 1 nếu đúng
giá trị 0 nếu sai.

a	b	!a	a && b	a||b
0	0	1	0	0
0	khác 0	1	0	1
khác 0	0	0	0	1
khác 0	khác 0	0	1	1

Ví dụ 1:
(4 > 5) && (6 < 8)
0 && 1
Kết quả 0

Ví dụ 2:
(4 > 5) || (6 < 8)
0 || 1
Kết quả 1

Ví dụ 3:
!((4 > 5) && (6 < 8))
!(0 && 1)
!0
Kết quả 1

8. Các phép toán quan hệ logic

- Các phép toán a && b và a || b luôn tính giá trị từ trái sang phải, a được tính trước tiên:
  - Nếu a là 0 ➔ a && b sai ➔không tính b.
  - Nếu a là khác 0 ➔ a || b đúng ➔ không tính b.

➔ Các phép toán đơn giản viết trước, phức tạp viết sau:

Ví dụ:

(x > y)	&&	(x + 2*z > t)	cách viết tốt
(x + 2*z > t)	&&	(x > y)	cách viết tồi

9. Các phép toán bitwise (CNTT)

- Phép toán từng cặp bit (bitwise) là phép toán thực hiện trên từng cặp bit của những số hạng tham gia phép toán.
- Các phép toán bitwise này gồm các phép toán sau:
  - & bitwise AND
  - ^ bitwise exclusive OR
  - | bitwise inclusive OR
- Khi thực hiện phép toán bitwise, hai số hạng của phép toán phải là số nguyên.

9. Các phép toán bitwise (CNTT)

Giá trị bit trong		Kết quả của phép toán		
x	y	x & y	x ^ y	x | y
0	0	0	0	0
0	1	0	1	1
1	0	0	1	1
1	1	1	0	1

Ví dụ các số nguyên biểu diễn bằng bằng 1 BYTE

$203_{10} = 11001011_2$

$93_{10} = 01011101_2$

Kết quả

$203 \mid 93 = 11011111_2 = 223_{10}$

$203 \,\&\, 93 = 01001001_2 = 73_{10}$

Nhận xét:
số chẵn là số có bit cuối cùng là 0
Số lẽ là số có bit cuối cùng là 1
➔Kiểm tra một số có là chẵn hay lẻ có thể sử dụng phép toán & bitwise

10. Phép toán dịch bit (CNTT)

- Phép toán dịch bit (Bitwise shift operators) gồm có phép toán dịch trái << và dịch phải >>:

x << y	Giá trị của x được dịch sang bên trái y bit, số 0 được bổ sung vào bên phải, tương đương với câu lệnh $x*2^y$
x >> y	Giá trị của x được dịch sang phải y bit, tương đương với câu lệnh $x/2^y$.

có thể thay thế phép toán: a * 2, a * 4, a*8, a*16
bằng các phép toán : a << 1, a << 2, a << 3, a << 4
Ví dụ

$93_{10} = 01011101_2$

93 >> 2 cho kết quả là $010111_2 = 23_{10}$
93 << 3 cho kết quả là $01011101000_2 = 744_{10}$

10. Bài tập về nhà

1. Hãy cho biết giá trị của biểu thức sau

   (6 > 7) && 9

2. Viết chương trình nhập một dãy số nguyên gồm có 10 phần tử. In ra màn hình:
   1. Dãy số nguyên
   2. Tổng các phần tử có giá trị chẵn
   3. Tìm phần tử có giá trị lớn nhất trong dãy số