

Kiểu dữ liệu

Kiểu dữ liệu	Bộ nhớ	Phạm vi giá trị
char	1 byte	−127 ... 127
short char	1 byte	−127 ... 127
unsigned char	1 byte	0 ... 255
sign char	1 byte	−127 ... 127
short int	2 byte	−32768 ... 32767
unsigned short int	2 byte	0 ... 65535
signed short int	2 byte	−32768 ... 32767
int	4 byte	−2147483648 ... 2147483648
unsigned int	4 byte	0 ... 4294967295
signed int	4 byte	−2147483648 ... 2147483648
long long	8 byte	−9223372036854775808 ... 9223372036854775808
unsigned long long	8 byte	0 ... 18446744073709551615
signed long long	8 byte	−9223372036854775808 ... 9223372036854775808
float (âm)	4 byte	$-(3,40282 \times 10^{38}) \dots -(1,17549 \times 10^{-38})$
float (dương)	4 byte	$1,17549 \times 10^{-38} \dots 3,40282 \times 10^{38}$
double (âm)	8 byte	$-(1,79769 \times 10^{308}) \dots -(2,22507 \times 10^{-308})$
double (dương)	8 byte	$2,22507 \times 10^{-308} \dots 1,79769 \times 10^{308}$

Hằng số

'\n' : biểu thị ký tự xuống hàng (cũng tương đương với endl)

'\t' : ký tự tab

Khai báo

#define tên_hằng giá_trị_hằng

hoặc:

const <kiểu dữ liệu> tên_hằng = giá_trị_hằng;

Ví dụ:

```
#define MAXN 1000
#define PI 3.14
const int MAXN = 1000;
const double PI = 3.14;
```

Lưu ý: tên hằng nên viết in hoa ví dụ: MAXN, PI

BIẾN

Khai báo không khởi tạo

<kiểu dữ liệu> <tên_biến_1>;

<kiểu dữ liệu> <tên_biến_2>;

<kiểu dữ liệu> <tên_biến_3>;

Nhiều biến cùng kiểu có thể được khai báo trên cùng một hàng:

<kiểu dữ liệu> <tên_biến_1>, <tên_biến_2>, <tên_biến_3>;

Ví dụ:

```
int n, m;  
double x;
```

Khai báo có khởi tạo

Trong câu lệnh khai báo, các biến có thể được gán ngay giá trị ban đầu bởi phép toán gán (=) theo cú pháp:

<kiểu dữ liệu> <tên_biến_1> = <gt_2>;

<kiểu dữ liệu> <tên_biến_1> = <gt_2>;

<kiểu dữ liệu> <tên_biến_1> = <gt_3>;

<kiểu dữ liệu> <tên_biến_1> = <gt_1>, <tên_biến_2> = <gt_2>, <tên_biến_3> = <gt_3>;

Trong đó các giá trị gt_1, gt_2, gt_3 có thể là các hằng, biến hoặc biểu thức.

Nhiều biến cùng kiểu có thể khai báo trên một hàng, các biến viết cách nhau dấu phẩy, dấu phẩy phải gõ sát từ đứng trước nó và sau nó là một dấu cách.

Ví dụ:

```
int i = 2, j , k = (i + 5) * 4;  
double eps = 1.0e-6;
```

Lưu ý: cách khai báo sau đây là sai

int n = m = 3 là sai mà phải khai báo như sau:

```
int n, m;
```

```
n = m = 3;
```

NHẬP - XUẤT

Nhập dữ liệu từ bàn phím

Để nhập dữ liệu vào cho các biến có tên bien_1, bien_2, bien_3 chúng ta sử dụng câu lệnh:

cin >> bien_1;

cin >> bien_2;

cin >> bien_3;

hoặc:

cin >> bien_1 >> bien_2 >> bien_3;

Ví dụ: cần nhập độ dài hai cạnh của một hình chữ nhật, trong đó cạnh dài được qui ước bằng tên biến cd và chiều rộng được qui ước bởi tên biến cr. Câu lệnh nhập sẽ như sau:

```
cin >> cd >> cr ;
```

Ví dụ: Nếu nhập vào: 5 3 thì chương trình sẽ gán giá trị 5 cho biến cd và 3 cho biến cr.

Chú ý: khi nhập giữa số 5 và 3 là dấu cách hoặc enter.

Xuất dữ liệu ra màn hình

Để xuất/in giá trị của các biểu thức ra màn hình ta dùng câu lệnh sau:

```
cout << bt_1;
```

```
cout << bt_2;
```

```
cout << bt_3;
```

hoặc:

```
cout << bt_1 << bt_2 << bt_3;
```

Ví dụ: Để in câu "Chiều dài là " và số 5 và tiếp theo là chữ " met", ta có thể sử dụng 3 lệnh sau đây:

```
cout << "Chiều dài là " ;
```

```
cout << 5 ;
```

```
cout << " met";
```

hoặc có thể chỉ bằng 1 lệnh:

```
cout << "Chiều dài là 5 met" ;
```

Trường hợp chưa biết giá trị cụ thể của chiều dài, chỉ biết hiện tại giá trị này đã được lưu trong biến `cd` và ta cần biết giá trị này là bao nhiêu thì có thể sử dụng câu lệnh in ra màn hình.

```
cout << "Chiều dài là " << cd << " met" ;
```

Ví dụ nếu `cd` được nhập là 5 thì lệnh trên sẽ in câu "Chiều dài là 5 met".

Một giá trị cần in không chỉ là một biến như `cd`, `cr`, ... mà còn có thể là một biểu thức, điều này cho phép ta dễ dàng yêu cầu máy xuất ra diện tích và chu vi của hình chữ nhật khi đã biết `cd` và `cr` bằng các câu lệnh sau:

```
cout << "Diện tích = " << cd * cr ;
```

```
cout << "Chu vi = " << 2 * (cd + cr) ;
```

hoặc gộp tất cả thành 1 câu lệnh:

```
cout << "Diện tích = " << cd * cr << '\n' << " Chu vi = " << 2 * (cd + cr) ;
```

Ký tự `'\n'` ký hiệu cho ký tự xuống hàng (ta cũng có thể sử dụng `cout << endl`), khi gặp ký tự này chương trình sẽ in các phần tiếp theo ở đầu hàng kế tiếp. Do đó kết quả của câu lệnh trên là 2 hàng sau đây trên màn hình:

Ví dụ: `cd = 5`, `cr = 3` thì:

```
Diện tích = 15
```

```
Chu vi = 16
```

Ở đây 15 và 16 lần lượt là các giá trị mà máy tính được từ các biểu thức `cd * cr` và `2 * (cd + cr)` trong câu lệnh in ở trên.