

## §7. CÁC THỦ TỤC CHUẨN VÀO/RA ĐƠN GIẢN

---

Để khởi tạo giá trị ban đầu cho biến, ta có thể dùng lệnh gán để gán một giá trị cho biến. Như vậy, mỗi chương trình luôn làm việc với một bộ dữ liệu vào. Để chương trình có thể làm việc với nhiều bộ dữ liệu vào khác nhau, thư viện của các ngôn ngữ lập trình cung cấp một số chương trình dùng để đưa dữ liệu vào và đưa dữ liệu ra.

Những chương trình đưa dữ liệu vào cho phép đưa dữ liệu từ bàn phím hoặc từ đĩa vào gán cho các biến, làm cho chương trình trở nên linh hoạt, có thể tính toán với nhiều bộ dữ liệu đầu vào khác nhau. Kết quả tính toán được lưu trữ tạm thời trong bộ nhớ. Những chương trình đưa dữ liệu ra dùng để đưa các kết quả này ra màn hình, in ra giấy hoặc lưu trên đĩa.

Các chương trình đưa dữ liệu vào và ra đó được gọi chung là các *thủ tục chuẩn vào/ra đơn giản*.

Trong phần này, ta sẽ xét các *thủ tục chuẩn vào/ra đơn giản* của Pascal để nhập dữ liệu vào từ bàn phím và đưa thông tin ra màn hình.

### 1. Nhập dữ liệu vào từ bàn phím

Việc nhập dữ liệu từ bàn phím được thực hiện bằng thủ tục chuẩn:

```
read(<danh sách biến vào>);
```

hoặc

```
readln(<danh sách biến vào>);
```

trong đó *danh sách biến vào* là *một* hoặc *nhiều* tên biến đơn (trừ biến kiểu boolean). Trong trường hợp nhiều biến thì các tên biến được viết cách nhau bởi *dấu phẩy*.

#### **Ví dụ**

```
read(N);  
readln(a, b, c);
```

Lệnh thứ nhất để nhập một giá trị từ bàn phím và gán giá trị đó cho biến *N*. Lệnh thứ hai dùng để nhập lần lượt ba giá trị từ bàn phím và gán các giá trị đó tương ứng cho ba biến *a*, *b* và *c*.

Khi nhập giá trị cho nhiều biến, những giá trị này được gõ cách nhau bởi ít nhất một dấu cách hoặc kí tự xuống dòng (nhấn phím **Enter**). Các giá trị ứng với biến nguyên phải được biểu diễn dưới dạng số nguyên (không có dấu chấm thập phân). Các giá trị ứng với biến thực có thể được nhập dưới dạng số nguyên, số thực dạng thông thường hoặc số thực dạng dấu phẩy động.

Ví dụ, để nhập các giá trị 1, -5 và 6 cho các biến thực  $a$ ,  $b$ ,  $c$  trong thủ tục thứ hai trong ví dụ trên, có thể gõ:

```
1 -5 6 rồi nhấn phím Enter  
hoặc
```

```
1.0 -5 rồi nhấn phím Enter  
6 rồi nhấn phím Enter.
```

## 2. Đưa dữ liệu ra màn hình

Để đưa dữ liệu ra màn hình, Pascal cung cấp thủ tục chuẩn:

```
write(<danh sách kết quả ra>);
```

hoặc

```
writeln(<danh sách kết quả ra>);
```

trong đó, *danh sách kết quả ra* có thể là tên biến đơn, biểu thức hoặc hằng. Các hằng xâu thường được dùng để tách các kết quả hoặc đưa ra chú thích. Các thành phần trong kết quả ra được viết cách nhau bởi dấu phẩy.

Với thủ tục *write*, sau khi đưa các kết quả ra màn hình, con trỏ không chuyển xuống dòng tiếp theo. Với thủ tục *writeln*, sau khi đưa thông tin ra màn hình, con trỏ sẽ chuyển xuống đầu dòng tiếp theo.

### Ví dụ

Để nhập giá trị cho biến  $M$  từ bàn phím, người ta thường dùng cặp thủ tục:

```
write('Hay nhap gia tri M: ');  
readln(M);
```

Khi thực hiện các lệnh này, trên màn hình xuất hiện dòng thông báo:

```
Hay nhap gia tri M:
```

và con trỏ sẽ ở vị trí tiếp theo trên dòng, chờ ta gõ giá trị của  $M$ .

Để chương trình được sử dụng một cách tiện lợi, khi nhập giá trị từ bàn phím cho biến, ta nên có thêm xâu kí tự nhắc nhở việc nhập giá trị cho biến nào, kiểu dữ liệu gì,... Ví dụ, khi cần nhập một số nguyên dương  $N$  ( $N \leq 100$ ) từ bàn phím, ta có thể sử dụng cặp thủ tục sau:

```
write('Nhap so nguyen duong N <= 100: ');  
readln(N);
```

### Ví dụ

Sau đây là một chương trình hoàn chỉnh có sử dụng các thủ tục vào và ra.

```
program Vidu;  
var N: byte;  
begin  
  write(' Lop ban co bao nhieu nguoi? ');  
  readln(N);  
  writeln(' Vay la ban co ', N-1, ' nguoi ban trong lop. ');  
  write('Go ENTER de ket thuc chuong trinh. ');  
  readln  
end.
```

Thủ tục *readln* cuối cùng dùng để tạm dừng thực hiện chương trình cho người dùng có thể quan sát kết quả của chương trình đưa ra trên màn hình. Muốn chương trình chạy tiếp cần nhấn phím **Enter**.

- Chú ý:**
- Các thủ tục *readln* và *writeln* có thể không có tham số.
  - Trong thủ tục *write* hoặc *writeln*, sau mỗi kết quả ra (biến, hằng, biểu thức) có thể có quy cách ra. Quy cách ra có dạng:
    - Đối với kết quả thực:  
: <độ rộng>; <số chữ số phần thập phân>
    - Đối với các kết quả khác:  
: <độ rộng>
- trong đó, *độ rộng* và *số chữ số phần thập phân* là các hằng nguyên không âm.

### Ví dụ

```
writeln(N:5, x:6:2);  
write(i:3, j:4, a+b:8:3);
```

uuu 36 u 24.00
425 uu 56 uu 23.200

*Minh họa với  $N = 36$ ,  $x = 24$ ,  $i = 425$ ,  $j = 56$  và  $a + b = 23.2$*

Trong thủ tục thứ nhất, 5 vị trí kể từ vị trí con trỏ hiện thời được dành để đưa ra giá trị  $N$ . Nếu  $N$  có giá trị nguyên dưới 5 chữ số hoặc giá trị âm dưới 4 chữ số thì những vị trí đầu sẽ được điền dấu cách. Tiếp theo là 6 vị trí được dành để đưa ra  $x$  ra, trong đó 2 vị trí dành để đưa ra phần thập phân. Do phần nguyên và phần thập phân được cách nhau bởi dấu chấm nên còn lại 3 vị trí cho phần nguyên.

Trong thủ tục thứ hai,  $i$  được đưa ra trên 3 vị trí,  $j$  được đưa ra trên 4 vị trí và kết quả  $a + b$  được đưa ra trên 8 vị trí, trong đó có 3 vị trí dành cho phần thập phân.