

NỘI DUNG CHÍNH BUỔI 1

(1) Các kiểu dữ liệu cơ sở

- Kiểu dữ liệu là sự kết hợp của 2 thành phần: miền giá trị mà kiểu dữ liệu có thể lưu trữ và tập hợp các phép toán để thao tác dữ liệu.
- Có 4 kiểu dữ liệu cơ sở:
 - Kiểu số nguyên, kiểu số thực, kiểu luận lý và kiểu ký tự.

Kiểu dữ liệu	Số bits (n)	Miền giá trị (từ -2^{n-1} đến $2^{n-1}-1$) / (từ 0 đến 2^n-1)
char	8	[-128, 127]
unsigned char	8	[0, 255]
int	16 / 32	[-32.768, 32.767] / [-2.147.483.648, 2.147.483.647]
unsigned int	16 / 32	[0, 65.535] / [0, 4.294.967.295]
short	16	[-32.768, 32.767]
unsigned short	16	[0, 65.535]
long	32	[-2.147.483.648, 2.147.483.647]
unsigned long	32	[0, 4.294.967.295]
float	32	[3.4E-38, 3.4E+38]
double	64	[1.7E-308, 1.7E+308]
long double	80	[3.4E-4932, 3.4E+4932]

Các kiểu dữ liệu thường dùng:

- Số nguyên (**int**)
- Số thực (**float** : 4 bytes)
- Số thực (**double** : 8 bytes)
- Ký tự (**char**)
- Luận lý (**bool**)

(2) Biến

- Biến là một vùng nhớ được lưu tại một địa chỉ nào đó trong bộ nhớ máy tính.
- Biến được đặt tên thông qua khai báo biến.
- Quy tắc đặt tên biến: **3 KHÔNG**
 - chứa khoảng trắng
 - chứa ký tự đặc biệt, từ khóa C đã dùng
 - bắt đầu là số
- Giá trị mà biến lưu trữ có thể bị thay đổi nhiều lần trong suốt quá trình chương trình thi hành.
- Cú pháp khai báo:

```
<kiểu biến> <Tên biến>;
<kiểu biến> <Tên biến 1>, <Tên biến 2>, ... , <Tên biến n>;
```

Ví dụ:

```
int i;
int j, k;
```

unsigned char dem;
float ketqua, delta;

(3) Hàm nhập dữ liệu

- Công dụng: lấy dữ liệu do người dùng nhập vào từ bàn phím và gán cho biến lưu trữ.
- Thư viện hàm: **stdio.h**
- Cú pháp:

```
scanf("đặc tả biến", địa chỉ biến);
```

○ Trong đó:

▪ **Đặc tả biến:**

- %d : số nguyên
- %f : số thực kiểu float
- %lf : số thực kiểu double
- %c : ký tự
- %s : chuỗi ký tự

▪ **Địa chỉ biến:**

- & tên_biến

(4) Hàm xuất dữ liệu

- Công dụng: hiển thị ra màn hình chuỗi văn bản, ký tự điều khiển, giá trị của 1 hoặc nhiều biến.
- Thư viện hàm: **stdio.h**
- Cú pháp:

```
printf("chuỗi_văn_bản ký_tự_điều_khiển đặc_tả_biến", tên_biến);
```

○ Trong đó:

▪ **Ký tự điều khiển:**

- \a : tiếng chuông
- \n : xuống dòng
- \t : dấu tab
- \\ : in dấu \
- \" : in dấu "

▪ **Đặc tả biến:**

- %d : số nguyên
- %f : số thực kiểu float
- %lf : số thực kiểu double
- %c : ký tự
- %s : chuỗi ký tự

- **Ví dụ 1: Hiển thị bảng thông tin họ tên, lớp như sau:**

```
printf("*****\n"); //In ra màn hình dòng dấu * và xuống dòng (\n)
printf("* Ho ten: TRAN VAN BINH *\n"); //In ra màn hình thông tin họ tên và xuống dòng (\n)
printf("* Lớp: KTLT C *\n"); //In ra màn hình thông tin lớp học và xuống dòng (\n)
printf("*****\n"); //In ra màn hình dòng dấu * và xuống dòng (\n)
```

- **Ví dụ 2:**

```
int i; //Khai báo biến i kiểu số nguyên
printf("Nhap so nguyen: "); /*Hiện thông báo trên màn hình yêu cầu người dùng nhập số nguyên*/
scanf("%d",&i); /*Chờ nhận giá trị số nguyên do người dùng nhập vào và gán cho biến i lưu trữ giá trị số nguyên này*/
printf("Gia tri so nguyen vua nhap vao la %d\n",i); /*Hiện thông báo trên màn hình giá trị của số nguyên mà người dùng vừa nhập vào*/
```

- **Ví dụ 3:**

```
int i,j; //Khai báo 2 biến i, j kiểu số nguyên
printf("Nhap 2 so nguyen: "); /*Hiện thông báo trên màn hình yêu cầu người dùng nhập 2 số nguyên*/
scanf("%d%d",&i,&j); /*Chờ nhận 2 giá trị số nguyên do người dùng nhập vào và lần lượt gán cho biến i, j lưu trữ giá trị 2 số nguyên này => Khi nhập liệu, người dùng có thể gõ 2 giá trị số nguyên liên tiếp, cách nhau bởi khoảng trắng và nhấn Enter để kết thúc nhập liệu, hoặc gõ số nguyên thứ nhất, nhấn Enter, rồi gõ số nguyên thứ 2, nhấn Enter*/
printf("Gia tri 2 so nguyen vua nhap vao la %d\t%d\n",i,j); /*Hiện thông báo trên màn hình giá trị của 2 số nguyên mà người dùng vừa nhập vào*/
```

- **Ví dụ 4:**

```
int i,j; //Khai báo 2 biến i, j kiểu số nguyên
double r; //Khai báo biến r kiểu số thực
printf("Nhap 2 so nguyen va 1 so thuc: "); /*Hiện thông báo trên màn hình yêu cầu người dùng nhập 2 số nguyên và 1 số thực*/
scanf("%d%d%lf",&i,&j,&r); /*Chờ nhận 2 giá trị số nguyên và 1 số thực do người dùng nhập vào và lần lượt gán cho biến i, j, r lưu trữ giá trị 3 số này => Khi nhập liệu, người dùng có thể gõ 2 giá trị số nguyên và 1 số thực liên tiếp, cách nhau bởi khoảng trắng và nhấn Enter để kết thúc nhập liệu, hoặc gõ số nguyên thứ nhất, nhấn Enter, gõ số nguyên thứ 2, nhấn Enter, rồi gõ số thực thứ 3, nhấn Enter */
printf("Gia tri 3 so vua nhap vao la %d\t%d\t%lf\n",i,j); /*Hiện thông báo trên màn hình giá trị của 3 số mà người dùng vừa nhập vào*/
```

- **Ví dụ 5:**

```
char kt; //Khai báo biến kt kiểu ký tự
int i; //Khai báo biến i kiểu số nguyên
printf("Nhap ky tu: "); /*Hiện thông báo trên màn hình yêu cầu người dùng nhập 1 ký tự*/
scanf("%c",&kt); /*Chờ nhận ký tự do người dùng nhập vào và gán cho biến kt lưu trữ giá trị ký tự này*/
printf("Nhap so nguyen: "); /*Hiện thông báo trên màn hình yêu cầu người dùng nhập số nguyên*/
scanf("%d",&i); /*Chờ nhận giá trị số nguyên do người dùng nhập vào và gán cho biến i lưu trữ giá trị số nguyên này*/
printf("Ky tu vua nhap vao la %c, gia tri so nguyen vua nhap vao la %d\n",kt,i); /*Hiện thông báo trên màn hình ký tự và giá trị của số nguyên mà người dùng vừa nhập vào*/
```

• **Ví dụ 6: tương tự Ví dụ 5 nhưng đổi thứ tự nhập số nguyên trước rồi nhập ký tự sau**

```
int i; //Khai báo biến i kiểu số nguyên
char kt; //Khai báo biến kt kiểu ký tự
printf("Nhap so nguyen: "); /*Hiện thông báo trên màn hình yêu cầu người dùng nhập số nguyên*/
scanf("%d",&i); /*Chờ nhận giá trị số nguyên do người dùng nhập vào và gán cho biến i lưu trữ giá trị số nguyên này*/
printf("Nhap ky tu: "); /*Hiện thông báo trên màn hình yêu cầu người dùng nhập 1 ký tự*/
scanf("%c",&kt); /*Chờ nhận ký tự do người dùng nhập vào và gán cho biến kt lưu trữ giá trị ký tự này*/
printf("Ky tu vua nhap vao la %c, gia tri so nguyen vua nhap vao la %d\n",kt,i); /*Hiện thông báo trên màn hình ký tự và giá trị của số nguyên mà người dùng vừa nhập vào*/
```

⇒ Kết quả là chương trình chỉ cho nhập số nguyên, không chờ người dùng nhập ký tự mà sẽ in thông báo trên màn hình giá trị số nguyên, còn ký tự vừa nhập vào là Enter.

Lý do:

- **stdin** là một vùng nhớ đệm dùng để lưu dữ liệu từ bàn phím.
- Hàm **scanf** nhận dữ liệu từ **stdin**. Nếu trên **stdin** không đủ dữ liệu theo yêu cầu của hàm **scanf**, thì máy tạm dừng để người dùng nhập liệu (đưa thêm dữ liệu từ bàn phím lên stdin cho đến khi nhấn phím Enter). Ngược lại, nếu trên **stdin** có đủ dữ liệu, thì hàm **scanf** sẽ nhận một phần dữ liệu mà chúng yêu cầu. Phần dữ liệu còn lại (chưa được nhận) vẫn ở trên **stdin**.
- Như vậy, khi nhập giá trị số nguyên, người dùng gõ giá trị số nguyên và nhấn phím **Enter** để kết thúc việc nhập số. Do đó, hàm **scanf** sẽ nhận giá trị số nguyên và để lại ký tự **Enter** trên **stdin**. Ký tự **Enter** này sẽ làm trôi hàm **scanf** chờ nhập ký tự sau đó. Để không bị trôi (tức là chương trình tạm dừng để chờ người dùng nhập ký tự) thì phải khử ký tự **Enter** trong **stdin** trước khi gọi hàm **scanf** bằng cách dùng hàm **fflush(stdin)** hoặc **flushall()** để làm sạch **stdin**.

```
int i; //Khai báo biến i kiểu số nguyên
char kt; //Khai báo biến kt kiểu ký tự
printf("Nhap so nguyen: "); /*Hiện thông báo trên màn hình yêu cầu người dùng nhập số nguyên*/
scanf("%d",&i); /*Chờ nhận giá trị số nguyên do người dùng nhập vào và gán cho biến i lưu trữ giá trị số nguyên này*/
fflush(stdin); //Hoặc hàm flushall();
printf("Nhap ky tu: "); /*Hiện thông báo trên màn hình yêu cầu người dùng nhập 1 ký tự*/
scanf("%c",&kt); /*Chờ nhận ký tự do người dùng nhập vào và gán cho biến kt lưu trữ giá trị ký tự này*/
printf("Ky tu vua nhap vao la %c, gia tri so nguyen vua nhap vao la %d\n",kt,i); /*Hiện thông báo trên màn hình ký tự và giá trị của số nguyên mà người dùng vừa nhập vào*/
```

Nếu **nhập ký tự trước, nhập số sau** thì không cần làm sạch **stdin**.

Nếu **nhập số trước, nhập ký tự sau** thì phải làm sạch **stdin** bằng cách dùng hàm **fflush(stdin)** hoặc **flushall()**.

- **Ví dụ 7:**

```
char hoten[50]; //Khai báo biến hoten kiểu chuỗi ký tự, biến hoten lưu trữ tối đa 49 ký tự vì có ký tự
kết thúc chuỗi quy ước là '\0' (ký tự có mã ASCII là 0) đặt ở cuối chuỗi.
printf("Nhập họ tên: "); /*Hiện thông báo trên màn hình yêu cầu người dùng nhập chuỗi họ tên*/
scanf("%s",&hoten); /*Chờ nhận chuỗi ký tự do người dùng nhập vào và gán cho biến hoten lưu trữ
chuỗi ký tự này*/
printf("Họ tên vừa nhập vào là %s\n",hoten); /*Hiện thông báo trên màn hình chuỗi ký tự họ tên mà
người dùng vừa nhập vào*/
```

- ⇒ Ví dụ người dùng nhập họ tên là TRAN VAN BINH. Kết quả là biến hoten chỉ lưu TRAN vì hàm scanf chỉ lấy các ký tự liên nhau không chứa khoảng trắng.
- ⇒ Sử dụng hàm **gets** để nhập một dãy ký tự từ stdin cho đến khi gặp Enter ('\n'). Ký tự '\n' bị loại khỏi stdin và dãy ký tự được bổ sung thêm ký tự kết thúc chuỗi '\0'.
- ⇒ Hàm **puts** xuất giá trị một chuỗi riêng trên 1 dòng.

```
char hoten[50]; //Khai báo biến hoten kiểu chuỗi ký tự, biến hoten lưu trữ tối đa 49 ký tự vì có ký tự
kết thúc chuỗi quy ước là '\0' (ký tự có mã ASCII là 0) đặt ở cuối chuỗi.
printf("Nhập họ tên: "); /*Hiện thông báo trên màn hình yêu cầu người dùng nhập chuỗi họ tên*/
gets(hoten); /*Chờ nhận chuỗi ký tự do người dùng nhập vào và gán cho biến hoten lưu trữ chuỗi ký
tự này*/
printf("Họ tên vừa nhập vào là %s\n",hoten); /*Hiện thông báo trên màn hình chuỗi ký tự họ tên mà
người dùng vừa nhập vào*/
//Hoặc dùng hàm puts như sau:
puts("Họ tên vừa nhập vào là ");
puts(hoten);
```