Kiểu dữ liệu

huydq@soict.hust.edu.vn

Biến

- Trong C, 1 biến phải được khai báo bởi người sử dụng với 1 tên định danh được đặt theo quy tắc:
  - VD: tong, dem, _abc, i, j, n
- Biến không chỉ được khai báo với tên mà còn phải có kiểu (biến của C luôn là biển có kiểu).
  - VD:

```
int i,j;
char ch;
```

đâu là biến?

```
#include <stdio.h>
int main()
{
  int tong=0, dem=0, sopt, sosau;

  printf(“So phan tu trong day so:");
  scanf("%d", &sopt);

  while (dem < sopt)
  {
    scanf("%d", &sosau);
    tong += sosau;
    dem++;
  }

  printf(“Tong la %d¥n",tong);
  return 0;
}
```

Kết quả

```
#include <stdio.h>
int main()
{
  int tong=0, dem=0, sopt, sosau;

  printf("So phan tu trong day so:");
  scanf("%d", &sopt);

  while (dem < sopt)
  {
    scanf("%d", &sosau);
    tong += sosau;
    dem++;
  }

  printf("Tong la %d¥n",tong);
  return 0;
}
```

Các kiểu cơ bản

- Mọi biến trong C đều dùng để lưu trữ con số
  - char : thường dùng lưu 1 số biểu diễn kí tự
  - short : lưu 1 số nguyên nhỏ (thường 16-bits)
  - int : lưu 1 số nguyên chuẩn (thường 32-bits)
  - long : lưu 1 số nguyên lớn
  - float : lưu 1số thực
  - double : lưu 1 số thực với độ chính xác cao

Khai báo biến

- Các biến trong C cần được khai báo khai bắt đầu một hàm (ví dụ bắt đầu hàm main) theo cú pháp

  <kiểu biến> <danh sách tên biến>;

  VD: int i,j;

- Dùng dấu = để khởi tạo giá trị ban đầu cho 1 biến

  int tong=0, dem=0

Số nguyên

- Số nguyên được lưu trữ dưới dạng số nhị phân trong bộ nhớ máy tính
  - VD: 10100110 là số mấy dạng thập phân?
- Có nhiều kiểu số nguyên khác nhau trong C. Mỗi kiểu quy định có kích thước lưu trữ khác nhau từ thấp đến cao như sau: char, short, int, long.
- để khai báo 1 biến nguyên có dấu hay không dấu, dùng từ khóa: unsigned và signed.
  - VD: signed int i; unsigned int u;

Kích thước số nguyên

Type	Bits	Possible Values
char	8	-128 to 127
short	16	-32768 to 32767
unsigned short	16	0 to 65535
int	32	-2147483648 to 2147483647
long	32	-2147483648 to 2147483647
unsigned int	32	0 to 4294967295
long long	64	-9e18 to + 8e18

Cần chú ý tới kích thước của các loại số để tránh tràn số trong các phép tính

Tìm giá trị phép tính

```
unsigned char a=128;
unsigned char b=128;
unsigned char c = a + b;
```

- Cho biết c có giá trị là bao nhiêu?

Hằng số nguyên

- để lưu các giá trị vào 1 biến ta cần có cách biểu diễn giá trị dưới dạng 1 hằng số.
- Biểu diễn dưới dạng số thập phân
  - VD: 123456, -123456
- Biểu diễn dưới dạng số hexa (với tiền tố 0x)
  - VD: 0x12AB, 0xFFFF
- Biểu diễn dưới dạng số bát phân (bắt đầu số bằng số 0)
  - VD: 0123456
- Cần chú ý hai giá trị 123456 và 0123456 là hai giá trị hoàn toàn khác nhau.

Ví dụ

```
short i=0,j=0;
short a = 0xFFFF;
int x,y;
x = y = 123456;
char k = 0xFF;
```

- Hãy cho biết giá trị của biến k? (biết char là biến nguyên có dấu kích thước 1 byte).

Số thực

- Cũng có 2 kiểu số thực "ngắn" và "dài" trong ngôn ngữ C tương ứng là: float và double
- Mọi hàm toán học trong C đều thực hiện trên số thực kiểu double.
- Chỉ nên sử dụng float khi cần tiết kiệm không gian bộ nhớ và bạn chỉ có các số thực giá trị nhỏ.

Kích thước số thực

Type	Bits	Possible Values
float	32	+/- 10E-37 to +/- 10E38
double	64	+/- 10E-307 to +/- 10E308

- Bản chất của số thực là được lưu trong bộ nhớ dưới dạng số nhị phân, do vậy nó vẫn luôn có tính rời rạc mà không thể có tính liên tục như 1 số thực trong tự nhiên.
- Do vậy khi ta có 1 giá trị thực bất kì trong tự nhiên là x, thì máy tính sẽ tìm 1 giá trị thực biểu diễn gần đúng nhất với x để lưu trữ nó.
- Chính vì vậy mọi phép tính số thực trong máy tính chỉ là phép tính "gần đúng". Khi ta chọn loại số thực có kích thước biểu diễn càng lớn thì các phép tính có độ chính xác càng cao.

Hằng số thực

- Sử dụng dấu chấm
  - VD: 123.456, -123.456
- Sử dụng kí pháp khoa học (E)
  - VD: 12.456e-2
- Ví dụ:

```
double x,y,z; x = 0.1;
y = 2.456E5;
z = 0;
```

Kí tự cũng là số!

- Các kí tự được lưu trữ như là một số nguyên nhỏ (1 byte) trong bộ nhớ
- để lưu trữ kí tự ta dùng kiểu char
- Mỗi một kí tự tương đương với một số duy nhất trong bảng mã ASCII.
- Có hai cách biểu diễn hằng kí tự trong C là kí tự trong cặp nháy đơn hoặc số mã ASCII của kí tự.
- Ví dụ kí tự 'A' có vị trí 65 trong bảng mã ASCII. Do đó hai khai báo sau là tương đương:
  - char c = 'A';
  - char c = 65;

ASCII value	Character	Control character	ASCII value	Character	ASCII value	Character	ASCII value	Character
0	(null)	NUL	32	(space)	64	@	96	
1	☺	SOH	33	!	65	A	97	a
2	☻	STX	34	"	66	B	98	b
3	♥	ETX	35	#	67	C	99	c
4	♦	EOT	36	$	68	D	100	d
5	♣	ENQ	37	%	69	E	101	e
6	♠	ACK	38	&	70	F	102	f
7	(beep)	BEL	39	'	71	G	103	g
8	◘	BS	40	(	72	H	104	h
9	(tab)	HT	41	)	73	I	105	i
10	(line feed)	LF	42	*	74	J	106	j
11	(home)	VT	43	+	75	K	107	k
12	(form feed)	FF	44	,	76	L	108	l
13	(carriage return)	CR	45	-	77	M	109	m
14	♫	SO	46	.	78	N	110	n
15	☼	SI	47	/	79	O	111	o
16	►	DLE	48	0	80	P	112	p
17	◄	DC1	49	1	81	Q	113	q
18	↕	DC2	50	2	82	R	114	r
19	!!	DC3	51	3	83	S	115	s
20	π	DC4	52	4	84	T	116	t
21	§	NAK	53	5	85	U	117	u
22	▬	SYN	54	6	86	V	118	v
23	↨	ETB	55	7	87	W	119	w
24	↑	CAN	56	8	88	X	120	x
25	↓	EM	57	9	89	Y	121	y
26	→	SUB	58	:	90	Z	122	z
27	←	ESC	59	;	91	[	123	{
28	(cursor right)	FS	60	<	92	\	124	|
29	(cursor left)	GS	61	=	93	]	125	}
30	(cursor up)	RS	62	>	94	^	126	~
31	(cursor down)	US	63	?	95	_	127	⌂

Các kí tự điều khiển

- Trong bảng mã ASCII
  - các kí tự có mã từ 32 (kí tự trắng) là kí tự hiển thị,
  - 32 kí tự đầu tiên (mã 0 ->mã 31) được dụng làm kí tự điều khiển
  - VD: kí tự mã 13 điều khiển xuống dòng, mã 9 điều khiển tao 1 khoảng tab, …
- Các hằng kí tự điều khiển được biểu diễn bằng cách thêm tiền tố '\'. Ví dụ:
  - **'\n' Biểu Diễn kí tự mã 13 xuống dòng**
  - **'\t' Biểu diễn kí tự mã 9 khoảng tab**
  - **'\0' Biểu diễn kí tự nul mã 0**
  - **'\'' Biểu diễn kí tự '**
  - **'\\' Biểu diễn một kí tự **

Chuỗi kí tự

- Là 1 nhóm các kí tự kết hợp thành 1 chuỗi văn bản
- Các chuỗi kí tự được biểu diễn trong cặp nháy kép (cặp nháy đơn chỉ biểu diễn 1 kí tự).
- Ví dụ:
  - “Hello world!”
  - “Line1\nLine2\nLine3”
  - “I\’m a teacher”

Logic

- Trong C, mọi con số ngoài giá trị thông thường còn thể hiện 1 giá trị logic,
  - sai nếu số có giá trị = 0,
  - đúng nếu số có giá trị khác 0.
  - Thông thường trong lập trình 1 được dùng để biểu diễn giá trị đúng về logic.
- Ví dụ:

```
int i = 1;
if ( i )
{
   printf("giá trị đúng");
}
```

Khai báo const

- Một khai báo biến với tiền tố const chỉ ra rằng giá trị của biến luôn luôn được giữ cố định trong toàn chương trình.
- Ví dụ
  - const int i = 5;
  - const char c = ‘A’;
  - const float pi = 3.14;