ĐẠI HỌC BÁCH KHOA HÀ NỘI
VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

C PROGRAMMING LANGUAGE

TS. Đỗ Quốc Huy
Bộ môn Khoa Học Máy Tính
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 biến 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;
}
```

Khai báo biến

- Một biến phải được khai báo trước khi sử dụng
- Cú pháp khai báo:

  <KieuDuLieu> TenBien;
  <KieuDuLieu> TenBien1, …, TenBien_N;
- Ví dụ:

```
//Khai báo biến x là một số nguyên 4 byte có dấu
        int x;
//Khai báo các biến y, z là các số thực 4 byte
        float  y,z;
//Sau khi khai báo, có thể sử dụng
        x = 3; y = x + 1;
```

Khai báo biến

- Sau khi khai báo, biến chưa có giá trị xác định.
  - Biến cần được gán giá trị trước khi sử dụng
- C cho phép kết hợp khai báo và khởi tạo biến

KieuDuLieu TenBien = GiaTriBanDau;

KieuDuLieu Bien1 = GiaTri1, BienN = Gia_TriN;

- Ví dụ:

```
//Khai báo biến nguyên a và khởi tạo gia tri bằng 3
int a = 3;
//Khai báo biến thực x,y và khởi tạo giá tri bằng 5.0 và 7.6
float x = 5.0, y = 7.6;
```

Khai báo hằng (dùng macro)

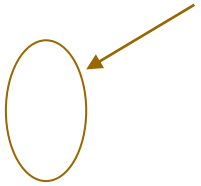
- Dùng chỉ thị #define
- Cú pháp:

  # define Tên_hằng Giá_trị

- Ví dụ:

  #define MAX_SINH_VIEN 50
  #define CNTT “Cong nghe thong tin”
  #define DIEM_CHUAN 23.5

Không có dấu ;



Khai báo hằng

- Dùng từ khóa const
- Cú pháp:

  const Kiểu Tên_hằng = giá_trị;

- Ví dụ:

```
const int  MAX_SINH_VIEN  = 50;
const char  CNTT[20] = “Cong nghe thong tin”;
const float  DIEM_CHUAN = 23.5;
```

Khai báo hằng

- Chú ý:
- Giá trị của các hằng phải được xác định ngay khi khai báo.
- #define là chỉ thị tiền xử lý
  - Khi chương trinh được biên dịch, <Tên_hằng> sẽ được thay thế bằng <Giá_trị>
  - Dễ đọc, dễ thay đổi
  - Dễ chuyển đổi giữa các nền tảng phần cứng hơn

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

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, kích thước lưu trữ khác nhau:

  char, short, int, long.
- Để khai báo 1 biến nguyên
  - có dấu, dùng từ khóa: signed
  - không dấu, dùng từ khóa: unsigned.
  - 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 2 giá trị 123456 và 0123456 là 2 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

Kiểu	Cỡ lưu trữ	Dãy giá trị	Độ chính xác
float	4 byte	1.2E-38 tới 3.4E+38	6 vị trí thập phân
double	8 byte	2.3E-308 tới 1.7E+308	15 vị trí thập phân
long double	10 byte	3.4E-4932 tới 1.1E+4932	19 vị trí thập phân

Kích thước số thực

- số thực được lưu trong bộ nhớ dưới dạng số nhị phân, → nó có tính rời rạc mà không thể có tính liên tục như số thực trong tự nhiên.

- → 1 giá trị thực bất kì trong tự nhiên là x, máy tính sẽ tìm 1 giá trị thực biểu diễn gần đúng nhất với x để lưu trữ nó.
  VD:0.666666666 lưu với kiểu float sẽ là 0.666667

- → mọi phép tính số thực trong máy tính chỉ là phép tính "gần đúng".

- Chọn loại số thực có kích thước biểu diễn càng lớn thì các phép tính có độ chính xác càng cao.

-> dùng kiểu double sẽ lưu được chính xác 0.666666666

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 1 số nguyên nhỏ (1 byte) trong bộ nhớ
- Để lưu trữ kí tự ta dùng kiểu char
- Mỗi kí tự tương đương với 1 số duy nhất trong bảng mã ASCII.
- Có 2 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 2 khai báo sau là tương đương:
  - char c = 'A';
  - char c = 65;

Hex	Dec	Char		Hex	Dec	Char	Hex	Dec	Char	Hex	Dec	Char
0x00	0	NULL	null	0x20	32	Space	0x40	64	@	0x60	96	`
0x01	1	SOH	Start of heading	0x21	33	!	0x41	65	A	0x61	97	a
0x02	2	STX	Start of text	0x22	34	"	0x42	66	B	0x62	98	b
0x03	3	ETX	End of text	0x23	35	#	0x43	67	C	0x63	99	c
0x04	4	EOT	End of transmission	0x24	36	$	0x44	68	D	0x64	100	d
0x05	5	ENQ	Enquiry	0x25	37	%	0x45	69	E	0x65	101	e
0x06	6	ACK	Acknowledge	0x26	38	&	0x46	70	F	0x66	102	f
0x07	7	BELL	Bell	0x27	39	'	0x47	71	G	0x67	103	g
0x08	8	BS	Backspace	0x28	40	(	0x48	72	H	0x68	104	h
0x09	9	TAB	Horizontal tab	0x29	41	)	0x49	73	I	0x69	105	i
0x0A	10	LF	New line	0x2A	42	*	0x4A	74	J	0x6A	106	j
0x0B	11	VT	Vertical tab	0x2B	43	+	0x4B	75	K	0x6B	107	k
0x0C	12	FF	Form Feed	0x2C	44	,	0x4C	76	L	0x6C	108	l
0x0D	13	CR	Carriage return	0x2D	45	-	0x4D	77	M	0x6D	109	m
0x0E	14	SO	Shift out	0x2E	46	.	0x4E	78	N	0x6E	110	n
0x0F	15	SI	Shift in	0x2F	47	/	0x4F	79	O	0x6F	111	o
0x10	16	DLE	Data link escape	0x30	48	0	0x50	80	P	0x70	112	p
0x11	17	DC1	Device control 1	0x31	49	1	0x51	81	Q	0x71	113	q
0x12	18	DC2	Device control 2	0x32	50	2	0x52	82	R	0x72	114	r
0x13	19	DC3	Device control 3	0x33	51	3	0x53	83	S	0x73	115	s
0x14	20	DC4	Device control 4	0x34	52	4	0x54	84	T	0x74	116	t
0x15	21	NAK	Negative ack	0x35	53	5	0x55	85	U	0x75	117	u
0x16	22	SYN	Synchronous idle	0x36	54	6	0x56	86	V	0x76	118	v
0x17	23	ETB	End transmission block	0x37	55	7	0x57	87	W	0x77	119	w
0x18	24	CAN	Cancel	0x38	56	8	0x58	88	X	0x78	120	x
0x19	25	EM	End of medium	0x39	57	9	0x59	89	Y	0x79	121	y
0x1A	26	SUB	Substitute	0x3A	58	:	0x5A	90	Z	0x7A	122	z
0x1B	27	FSC	Escape	0x3B	59	;	0x5B	91	[	0x7B	123	{
0x1C	28	FS	File separator	0x3C	60	<	0x5C	92	\	0x7C	124	|
0x1D	29	GS	Group separator	0x3D	61	=	0x5D	93	]	0x7D	125	}
0x1E	30	RS	Record separator	0x3E	62	>	0x5E	94	^	0x7E	126	~
0x1F	31	US	Unit separator	0x3F	63	?	0x5F	95	_	0x7F	127	DEL

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");
}
```

Chương trình ví dụ

```c
#include <stdio.h>
#include <limits.h>

int main()
{
        printf("Kich co luu tru cho so nguyen (int) la: %d \n", sizeof(int));
        return 0;
}
```

In giá trị lớn nhất và nhỏ nhất kiểu float

```
#include <stdio.h>
#include <float.h>

int main() {
        printf("Lop luu tru cho so thuc (float) la: %d \n", sizeof(float));
        printf("Gia tri so thuc duong nho nhat la: %E\n", FLT_MIN );
        printf("Gia tri so thuc duong lon nhat la: %E\n", FLT_MAX );
        printf("Do chinh xac: %d\n", FLT_DIG );
        return 0;
}
```

Các kiểu đơn

Tên kiểu	Ý nghĩa	Kích thước	Miền dữ liệu
char	Kí tự; Số nguyên có dấu	1 byte	-128 ÷ 127
short int	Số nguyên có dấu	2 byte	-32.768÷32.767
int	Số nguyên có dấu	2 hoặc **4 byte**	
long long int	Số nguyên có dấu	4 byte	-2,147,483,648 ÷ 2,147,483,647
float	Số thực dấu phẩy động, độ chính xác đơn	4 byte	± 3.4E-38 ÷ ± 3.4E+38
double	Số thực dấu phẩy động, độ chính xác kép	8 byte	±1.7E-308 ÷ ± 1.7E+308

Các kiểu kết hợp

Với số nguyên, thêm từ khóa unsigned để chỉ ra số không dấu

Kiểu dữ liệu	Ý nghĩa	Kích thước	Miền dữ liệu
unsigned char	Số nguyên không dấu	1 byte	0 ÷ 255
unsigned short	Số nguyên không dấu	2 byte	0÷65.535
unsigned int	Số nguyên không dấu	2 hoặc 4 byte	
unsigned long unsigned long int	Số nguyên không dấu	4 byte	0 ÷ 4,294,967,295
long double	Số thực dấu phẩy động,	10 byte	±3.4E-4932 ÷ ±1.1E+4932
void	Là kiểu đặc biệt, không có kích thước		