

C语言基本知识

陈 斌

目录

第一节 概述

第二节 基本知识

第三节 流程控制

第四节 指针与数组

第五节 函数

Hello, world!

例10-1 Hello, world!

```
#include <stdio.h>
/*预处理语句引入系统头文件, 可以使用系统提供的许多函数 */

int main(void)
//每个程序都必须而且只能包含一个main函数
{
    printf("hello, world!\n");
    return 0;
}
```

C程序的书写

- ❖ 每条语句以末尾的分号“;”表示结束;
- ❖ 每行通常写一条语句, 也可以写多条语句。如果语句太长, 可以拆开换行继续书写, 一般不用加续行标志。
- ❖ 注释语句以“/*”开头, 以“*/”结尾, 如果注释较长, 可以写作多行, 包含在“/*”与“*/”之间
- ❖ 可以用花括号{}把多个单条语句括起来, 组成一个复合语句。复合语句内的各条语句都必须以分号“;”结尾, 在括号“}”外不能加分号。

字符集

❖ 字符集:

- 52个字母 (C语言区分大小写)
- 10个阿拉伯数字
- 29个特殊字符
 - ! + - * / = % ' " #
 - { } [] < > ()
 - . , ; : ? ~ ^ \ |
 - _ &
- 4个格式符
 - 空格、水平制表符 (HT)、垂直制表符 (VT)、换页符 (FF)

❖ 其他字符只能放在注释语句、字符型常量、字符串型常量中

数据类型

❖ 数据类型:

- 基本类型:
 - 整型、字符、浮点 (单精度、双精度)、枚举
- 构造类型
 - 数组、结构体 (struct) 和联合 (union)
- 指针类型 (*)
- 空类型 (void)

数据类型

类型	字宽	范围	最小值	最大值	
整型	short	2	-32768~32767	SHRT_MIN	SHRT_MAX
	unsigned short	2	0~65535		USHRT_MAX
	int	4	-2147483648~2147483647	INT_MIN	INT_MAX
	unsigned int	4	0~4294967295		UINT_MAX
	long	4	-2147483648~2147483647	LONG_MIN	LONG_MAX
单精度实型	float	4	1.175494351e-038 ~3.402823466e+38	FLT_MIN	FLT_MAX
双精度实型	double	8	2.2250738585072014e-308 ~1.7976931348623158e+308	DBL_MIN	DBL_MAX
	long double	10	3.3621031431120935063e-4932 ~1.189731495357231765e+4932	LDBL_MIN	LDBL_MAX
字符型	char	1	-128~127	CHAR_MIN	CHAR_MAX
	signed char	1	-128~127	SCHAR_MIN	SCHAR_MAX
	unsigned char	1	0~255		UCHAR_MAX

数据类型

```
#include <stdio.h>
#include <stdlib.h>
#include <limits.h>
#include <float.h>
int main()
{
    printf("%d, %d\n", SHRT_MIN, SHRT_MAX);
    printf("%u\n", USHRT_MAX);
    printf("%d, %d\n", INT_MIN, INT_MAX);
    printf("%u\n", UINT_MAX);
    printf("%ld, %ld\n", LONG_MIN, LONG_MAX);
    printf("%lu\n", ULONG_MAX);
    printf("%d, %d\n", CHAR_MIN, CHAR_MAX);
    printf("%d, %d\n", SCHAR_MIN, SCHAR_MAX);
    printf("%d\n", UCHAR_MAX);
    printf("%.6e, %.6e\n", FLT_MIN, FLT_MAX);
    printf("%.12e, %.12e\n", DBL_MIN, DBL_MAX);
    printf("%.15Lf, %.15Lf\n", LDBL_MIN, LDBL_MAX); /*部分编译器不支持*/
    return 0;
}
```

例10-2 各种数据类型的最大、最小值

标识符与常量

❖ 标识符

- 由字母或下划线开头。由于C语言自带的库函数通常以下划线开头，因此标识符命名时不要以下划线开头
- 其它部分可以由字母、下划线或数字组成；
- 有效长度为32个字符；
- 区分字母大小写。

❖ 常量

- 整型常量、浮点常量、字符常量
- 枚举常量
- 符号常量

枚举常量

❖ 枚举常量：一个整型常量值的列表

- 如果一个变量存在几种可能的值，就可以被定义成为枚举类型。
- `enum` 常量名 {枚举名列表}
- 如果没有显式说明，枚举类型中第一个枚举名的值为0，第二个为1，依次类推。如果只指定了部分枚举名的值，那么未指定值的枚举名，其取值从最后一个指定值递增。

```
enum color {RED, ORIENT, YELLOW, GREEN, CYAN, BLUE, PURPLE}
/*RED的值为0, ORIENT的值为1, 依次类推*/
enum shape {CIRCLE = 2, SQUARE, RECTANGLE, TRIANGLE, OVAL}
/*CIRCLE的值为2, SQUARE的值为3, 依次类推*/
```

枚举常量

❖ 只能把枚举值赋予枚举变量，不能把元素的数值直接赋予枚举变量：

```
a = sum;
b = mon;
▪ 是正确的。而：
a = 0;
b = 1;
▪ 是错误的。
```

❖ 如一定要把数值赋予枚举变量，则必须用强制类型转换：

```
▪ a = (enum weekday)2;
```

❖ 其意义是将顺序号为2的枚举元素赋予枚举变量a，相当于：

```
▪ a = tue;
```

❖ 枚举元素不是字符常量也不是字符串常量，使用时不要加单、双引号。

枚举常量

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

enum week {Monday=1, Tuesday, Wednesday, Thursday, Friday, Saturday, Sunday};

int main()
{
    int i;

    enum week today, tomorrow;

    scanf("%d", &i);

    switch(i)
    {
        case 1: today = Monday; break;
        case 2: today = Tuesday; break;
        case 3: today = Wednesday; break;
        case 4: today = Thursday; break;
        case 5: today = Friday; break;
        case 6: today = Saturday; break;
        case 7: today = Sunday; break;
        default: break;
    }

    if (today == Sunday)
        tomorrow = Monday;
    else
        tomorrow = today + 1;

    printf("Tomorrow is ");

    switch(tomorrow)
    {
        case Monday: printf("Monday"); break;
        case Tuesday: printf("Tuesday"); break;
        case Wednesday: printf("Wednesday"); break;
        case Thursday: printf("Thursday"); break;
        case Friday: printf("Friday"); break;
        case Saturday: printf("Saturday"); break;
        case Sunday: printf("Friday"); break;
        default: break;
    }

    return 0;
}
```

符号常量

- ❖ 符号常量：用一个标识符来代表一个常量
- ❖ #define 宏名 常数
 - #define PI 3.14159265
 - #define E 2.71828
- ❖ 一旦完成定义后，符号常量的值在作用域内不能改变，也不能再被赋值
 - 宏定义的实质是用宏名代替字符串，只作简单的置换，不做语法检查；
 - 宏名一般习惯用大写字母表示，以与变量名相区别；
 - 宏定义不是C语句，无需在行末加分号。如果加了分号则会连分号一起置换。

符号常量

```
#include <stdio.h>

#define PI 3.14159265      /* 全局变量 */

int main()
{
    float r, a, v;

    scanf("%f", &r);

    a = 4.0 * PI * r * r;
    v = 4.0 / 3.0 * PI * pow(r, 3);

    printf("a = %f  b = %f\n", a, v);

    return 0;
}
```

变量及其初始化

- ❖ 变量：使用之前必须先定义其数据类型，未经声明的变量不能使用。变量声明通常放在函数起始处，位置在第一条可执行语句前面。
 - int i, j, k;
 - double temperature, pressure, timestep;
- ❖ 声明的同时可以对变量进行初始化
 - int i=1;
 - float x = 5.0;

变量的const修饰

- ❖ 所有变量的声明都可以用const修饰符，表示指定该变量的值不能被修改：
 - const double e = 2.71828;
 - const double pi = 3.14159265;
 - double const e = 2.71828;
 - double const pi = 3.14159265;

宏定义	const修饰变量
#define PI 3.14169265;	const double pi = 3.14159265;
定义宏，预编译时处理 只进行简单的字符替换，无类型检查	定义程序中的常量，编译时处理 常量修饰符，把变量定义为常量
作用范围在整个程序	只能在定义该变量的函数体内使用，除非定义的是全局变量

运算符

- ❖ 按运算的类别
 - 算术运算符
 - 关系运算符
 - 逻辑运算符
 - 位操作运算符
 - 赋值运算符
 - 条件运算符 (?和:)
 - 逗号运算符 (,)
 - 指针运算符 (*和&)
 - 求字节数运算符 (sizeof)
 - 特殊运算符 ((), [], ., ->)
- ❖ 按操作数的多少
 - 单目运算符 (一元)
 - 双目运算符 (二元)
 - 多目运算符 (多元)

混合运算与左右结合性

- ❖ 不同运算符混合计算时:
- ❖ 按优先符的优先级由高向低进行运算, 例如先乘除、后加减。
- ❖ 当一个操作数两侧的运算符优先级相同时, 按运算符的结合性所规定的结合方向处理。
 - 左结合性 (自左至右)
算术运算符: $a*b/c$
 - 右结合性 (自右至左)
赋值运算符: $a=b=c$
相当于: $b=c; a=b;$

运算符与左右结合性

运算符	优先级	结合性
() (小括号) [] (数组下标) . (结构成员) -> (指针型结构成员)	最高	从左至右
! (逻辑非) ~ (按位取反) ++ (自增) -- (自减) sizeof (求字节数) * (指针运算符) & (地址运算符) + (正号) - (负号)	↑	从右至左
* (乘) / (除) % (取余)		从左至右
+ (加) - (减)		从左至右
<< (左移) >> (右移)		从左至右
< (小于) <= (小于等于) > (大于) >= (大于等于)		从左至右
== (等于) != (不等于)		从左至右
& (按位与)		从左至右
^ (按位异或)		从左至右
(按位或)		从左至右
&& (逻辑与)		从左至右
(逻辑或)		从右至左
?: (问号表达式)		从右至左
= (赋值) += -= *= /= %= &= ^= = <<= >>= (复合赋值)		从左至右
, (逗号表达式)		最低

运算符

❖ 算术运算: + - * / % (取余) pow(x, y)

❖ 自增、自减运算符

- i++ (i--) 表示使用i后将i的值加1 (减1)
- ++i (--i) 表示将i的值加1 (减1)后再使用i

```
#include <stdio.h> //例10-3

int main()
{
    int i=10, j=10, m, n;

    m = (i++) + (i++);
    n = (++j) + (++j);
    printf("%d %d %d %d", m, n, i, j);

    return 0;
}
```

运算符

❖ 赋值运算符：具有右结合性

▪ $i=j=k=1;$

❖ 复合赋值运算符：

▪ 在赋值运算符“=”之前加上其他运算符可构成复合赋值运算符

▪ $i += 1$ 即 $i = i + 1$

▪ $i *= j+1$ 即 $i = i * (j+1)$

▪ $+=$ $-=$ $*=$ $/=$ $\%=$

▪ $<<=$ $>>=$ $\&=$ $\^=$ $|=$

能源与动力工程学院

Thank You !