

C语言考前复习

1.标识符

定义变量时，我们使用了诸如 a、abc、mn123 这样的名字，它们都是程序员自己起的，一般能够表达出变量的作用，这叫做标识符 (Identifier) 。

标识符就是程序员自己起的名字，除了变量名，后面还会讲到函数名、宏名、结构体名等，它们都是标识符。不过，名字也不能随便起，要遵守规范；C语言规定，标识符只能由字母 (A~Z, a~z)、数字 (0~9) 和下划线 (_) 组成，并且第一个字符必须是字母或下划线，不能是数字。

2.关键字

C共有32个关键字

关键字	说明
auto	声明自动变量
short	声明短整型变量或函数
int	声明整型变量或函数
long	声明长整型变量或函数
float	声明浮点型变量或函数
double	声明双精度变量或函数
char	声明字符型变量或函数
struct	声明结构体变量或函数
union	声明共用数据类型
enum	声明枚举类型
typedef	用以给数据类型取别名
const	声明只读变量
unsigned	声明无符号类型变量或函数
signed	声明有符号类型变量或函数
extern	声明变量是在其他文件正声明
register	声明寄存器变量
static	声明静态变量

volatile	说明变量在程序执行中可被隐含地改变
void	声明函数无返回值或无参数，声明无类型指针
if	条件语句
else	条件语句否定分支（与 if 连用）
switch	用于开关语句
case	开关语句分支
for	一种循环语句
do	循环语句的循环体
while	循环语句的循环条件
goto	无条件跳转语句
continue	结束当前循环，开始下一轮循环
break	跳出当前循环
default	开关语句中的“其他”分支
sizeof	计算数据类型长度
return	子程序返回语句（可以带参数，也可不带参数）循环条件

3.转义、输出控制

转义字符以\或者\x开头，以\开头表示后跟八进制形式的编码值，以\x开头表示后跟十六进制形式的编码值。对于转义字符来说，只能使用八进制或者十六进制。

字符 1、2、3、a、b、c 对应的 ASCII 码的八进制形式分别是 61、62、63、141、142、143，十六进制形式分别是 31、32、33、61、62、63。

转义字符的初衷是用于 ASCII 编码，所以它的取值范围有限：

八进制形式的转义字符最多后跟三个数字，也即\ddd，最大取值是\177；

十六进制形式的转义字符最多后跟两个数字，也即\xdd，最大取值是\7f。

超出范围的转义字符的行为是未定义的，有的编译器会将编码值直接输出，有的编译器会报错。

对于 ASCII 编码，0~31（十进制）范围内的字符为控制字符，它们都是看不见的，不能在显示器上显示，甚至无法从键盘输入，只能用转义字符的形式来表示。不过，

直接使用 ASCII 码记忆不方便，也不容易理解，所以，针对常用的控制字符，C语言又定义了简写方式，完整的列表如下：

转义字符	意义	ASCII码值（十进制）
\a	响铃(BEL)	007
\b	退格(BS)，将当前位置移到前一系列	008
\f	换页(FF)，将当前位置移到下页开头	012
\n	换行(LF)，将当前位置移到下一行开头	010
\r	回车(CR)，将当前位置移到本行开头	013
\t	水平制表(HT)	009
\v	垂直制表(VT)	011
\'	单引号	039
\"	双引号	034
\\	反斜杠	092

奇怪考点：ASCII码中000-002分别为null,start of head和start of text，都不会显示，所以'\000a'是在gcc看来是合法的，即'a'。

输出控制格式符的语法

syntax: %[flag][width][.precision][h|l]type

a) [h|l]type

type字段采用一个英文字母来表达数据类型与格式:

d	按十进制有符号整数形式输出
i	按十进制有符号整数形式输出(同d格式)
u	按十进制无符号数形式输出
o	按八进制无符号数形式输出
x	按十六进制无符号数形式输出，输出时使用小写字母 (a, b, c, d, e, f)
X	按十六进制无符号数形式输出，输出时使用大写字母(A, B, C, D, E, F)
f	按十进制小数形式输出浮点数，输出格式为:[-]ddd.dddddd（默认输出6位小数）

e	按十进制指数形式输出浮点数，输出格式为：[-]d.dddde[+/-]ddd,(e后面是指数)
E	按十进制指数形式输出浮点数，输出格式为：[-]d.dddE[+/-]ddd,(E后面是指数)。 用e和E格式输出浮点数时，输出的是科学记数法形式，即小数点前面的整数部分固定为1位整数
g	按十进制形式输出浮点数，自动选择f或e格式中，输出长度小的格式输出；g格式不输出无用的0。
G	按十进制形式输出浮点数，自动选择f或E格式中，输出长度小的格式输出；G格式不输出无用的0。
c	输出单个字符
s	输出字符串
p	输出void指针，输出格式是：WIN32编程模式，只有XXXXXXXX一种格式
%	输出字符%（%用于引导格式控制符，在格式串中输出%号时，必须采用格式%%）

前缀修饰符：

L: 输出long double类型表达式必加

l: 输出长整型表达式必加

h: 输出短整型表达式必加

b) [width]

width字段用来指定输出的数据项占用的字符列数，也称为输出域宽。缺省该字段，输出宽度按数据的实际位数输出；如果指定的输出宽度小于数据的实际位数，则突破域宽的限制，按实际位数输出；如果指定的域宽大于数据的实际位数，则默认在输出数据的左边输出空格，使输出的字符数等于列宽，也就是说，输出的数据在输出域中自动向右对齐。

width有以下三种情况：

n	一个非负整型常数，指定输出占用n列宽度。
0n	n为一个整型常数，输出占用n列，如果实际位数不足n列，数据前面补0，填满n列
*	输出域宽来自待输出表达式前面的一个整型表达式。

c) [.precesion]

有以下4种情况：

缺省	对于f e E格式, 表示小数点后输出6位小数,对于g G格式, 表示最多输出6位有效数字
.m	m为非负整型常数, 对于浮点格式, 指定小数点后面输出m位小数; s格式, 表示只输出字符串的前面m个字符
.0	对于f e E格式, 表示不输出小数点和小数, 按整数形式输出
.*	表示precision来自待输出表达式前面的一个整型表达式

示例

```
#include "iostream"
#include "stdio.h"
using namespace std;

int main()
{
    double a=12.578,b=0.2572;
    int n=9,m=4;
    printf("%.2f,%10.2E,%f,%e,%10g",a,a,a,a,a);
    printf("\n");
    printf("a=%.0f,a=%*.*f",a,n,m,a);
    printf("\n");
    printf("%.2f%%",b*100);
    printf("\n");
    printf("%s%6s%6.3s","ABCD", "ABCD", "ABCD");
    return 1;
}
```

```
C:\WINDOWS\system32\cmd.exe
12.58, 1.26E+001,12.578000,1.257800e+001, 12.578
a=13,a= 12.5780
25.72%
ABCD ABCD ABC请按任意键继续...
```

d) [flag]

缺省	输出正数时不输出正号;域宽大于数据实际位数时, 域中左边补空格, 数据靠右对齐;
+	输出正数时要输出正号;
-	域宽大于数据实际位数时, 域中右边补空格, 数据靠左对齐。
#	用于x X格式: 输出0x (格式x) 或0X (格式X) 。

输入控制符的语法

syntax: %[*][width][h||L]type

(1) [h||L]type

d	以十进制有符号整数形式转换输入数据
i	以十进制有符号整数形式转换输入数据(同d格式)
u	以十进制无符号整数形式转换输入数据
o	以八进制有符号整数形式转换输入数据
x X	以十六进制有符号整数形式转换输入数据(x和X等效)
e E f g G	以十进制浮点数形式转换输入数据, 输入数据时, 可以输入整型常量、小数形式实型常量或指数形式实型常量。
c	输入一个字符(可输入控制字符)
s	输入字符串(遇到第一个空格、TAB或换行符结束转换)

前缀修饰符:

L: 输入long double类型变量必加

l: 输入长整型变量或者double型变量必加

h: 输入短整型变量必加

(2) [width]

width字段用来指定输入数据的转换宽度, 它必须是一个十进制非负整型常量。width表示读入多少个字符就结束本数据项的转换。如果没有指定width, 则遇到空格、TAB键、回车/换行符、非法输入则结束数据项的转换(%c格式除外)。

float a;int b;double c;

scanf("%f%3x%lf",&a,&b,&c);

若输入为: -1.5A8059;

则a=-1.5, b=0xA80, c=59.0

(3) [*]号

表示数据输入项要按指定格式进行转换, 但不保存变量, 即该%没有对应的变量。

一般用%*c来吸收字符。

2、几点说明：

编译程序只是检查printf 函数的调用形式，不分析格式控制字符串，如果格式字符与输出项的类型不匹配，不进行类型转换。（输出的数为随机）

格式字符要用小写字母。

格式控制字符串中可以包含转义字符。

格式控制字符串中的普通字符则原样输出。

输出项的参数除了常数、变量外还可以是表达式、函数调用。

4.运算符优先级

优先级	运算符	名称或含义	使用形式	结合方向	说明
1	[]	数组下标	数组名[常量表达式]	左到右	
	()	圆括号	(表达式) /函数名(形参表)		
	.	成员选择 (对象)	对象.成员名		
	->	成员选择 (指针)	对象指针->成员名		
	++	后置自增运算符	++变量名		单目运算符
	--	后置自减运算符	--变量名		单目运算符
2	-	负号运算符	-表达式	右到左	单目运算符
	(类型)	强制类型转换	(数据类型)表达式		
	++	前置自增运算符	变量名++		单目运算符
	--	前置自减运算符	变量名--		单目运算符
	*	取值运算符	*指针变量		单目运算符
	&	取地址运算符	&变量名		单目运算符
	!	逻辑非运算符	!表达式		单目运算符
	~	按位取反运算符	~表达式		单目运算符
	sizeof	长度运算符	sizeof(表达式)		
3	/	除	表达式/表达式	左到右	双目运算符
	*	乘	表达式*表达式		双目运算符
	%	余数（取模）	整型表达式/整型表达式		双目运算符
4	+	加	表达式+表达式	左到右	双目运算符

	-	减	表达式-表达式		双目运算符
5	<<	左移	变量<<表达式	左到右	双目运算符
	>>	右移	变量>>表达式		双目运算符
6	>	大于	表达式>表达式	左到右	双目运算符
	>=	大于等于	表达式>=表达式		双目运算符
	<	小于	表达式<表达式		双目运算符
	<=	小于等于	表达式<=表达式		双目运算符
7	==	等于	表达式==表达式	左到右	双目运算符
	!=	不等于	表达式!= 表达式		双目运算符
8	&	按位与	表达式&表达式	左到右	双目运算符
9	^	按位异或	表达式^表达式	左到右	双目运算符
10		按位或	表达式 表达式	左到右	双目运算符
11	&&	逻辑与	表达式&&表达式	左到右	双目运算符
12		逻辑或	表达式 表达式	左到右	双目运算符
13	?:	条件运算符	表达式1? 表达式2: 表达式3	右到左	三目运算符
14	=	赋值运算符	变量=表达式	右到左	
	/=	除后赋值	变量/=表达式		
	=	乘后赋值	变量=表达式		
	%=	取模后赋值	变量%=表达式		
	+=	加后赋值	变量+=表达式		
	-=	减后赋值	变量-=表达式		
	<<=	左移后赋值	变量<<=表达式		
	>>=	右移后赋值	变量>>=表达式		
	&=	按位与后赋值	变量&=表达式		
	^=	按位异或后赋值	变量^=表达式		
	=	按位或后赋值	变量 =表达式		
15	,	逗号运算符	表达式,表达式,...	左到右	从左向右顺序运算