



黑马程序员  
itheima.com

——为莘莘学子改变命运而讲课  
为千万人少走弯路而著书！

iOS

# C语言重难点



传智播客 & 黑马程序员 联合出品

## 回答问题

1. C程序由什么组成？
2. 程序可以有多少个函数？
3. 一个程序主函数有多少个？
4. 没有主函数会出什么问题？能不能编译呢？
5. 函数不调用会不会执行？
6. 程序代码能不能直接运行？为什么？
7. 那怎样才能运行？说说步骤
8. 链接过程最主要的工作是什么？
9. 链接的命令是什么？编译呢？
10. 终端运行一个程序要注意什么问题？
11. 输出函数叫什么？怎么换行？能用这个函数的前提是什么？



黑马程序员  
itheima.com

——为莘莘学子改变命运而讲课  
为千万人少走弯路而著书！

iOS

# 数据类型

难易程度:★★★

重要程度:★★★★★



# C 语言数据类型



# 变量

## 一,是什么?

变量来源于数学,是计算机语言中能储存计算结果或能表示值抽象概念.

## 二,有什么用?

用来表示一个需要经常改变或者不确定的数据

## 怎么用?

变量类型 变量名; 如: `int num;`

1.赋值 `num=1;` 2.修改 `num=2;`

## 四,什么时候用?

1.需要定义一个经常改变或者不确定的数据.

## 五,有什么特点?

1.创建一个变量相当于在内存中占用了一块空间

2.这个空间的大小是固定的,但里面的数据可以改变

3.同一个作用域内不能有相同的变量名.

4.初始化时创建,在函数作用域执行完后失效,且内存空间释放.

# 变量的内存分析

1. 内存的基本单位的是byte(字节).每一块字节都有地址(可以理解为名字)
2. 变量创建后就占用了一片固定内存空间.
3. 不同类型的数据变量在不同编译环境下的占用内存空间大小是不一样的.

	16位编译器	32位编译器	64位编译器
char	1	1	1
int	2	4	4
float	4	4	4
double	8	8	8

4. 输出内存占用: `printf("char类型在当前环境占用内存字节数为:%lu\n", sizeof(char));`
5. 变量创建后会分配一个地址(就是其占用的第一个字节的地址)
6. 输出内存地址的方法: `int num; printf("num的地址是: %p\n", &num);`
7. 内存由地址从大到小寻址, 优先分配内存地址较大的字节给变量。
8. 在变量未经初始化赋值之前, 不要使用变量的值.(没有意义).

# scanf () 函数

## 一,是什么?

C语言下的scanf()函数是样式化输入函数.由C语言自带的库提供.

## 二,有什么用?

可以按用户指定的格式从键盘上把数据输入到指定的变量之中.

## 三,怎么用?

1. #include<stdio.h>//引入头文件.表示在这个文件里要用到里面的scanf ()函数.

2. scanf ("格式化字符串", 变量地址);如 int a;int b; scanf("%d, %d",&a,&b);

3. 格式化规定字符,以"%"开始,后跟一个或几个规定字符

常用:%d 整数      %f 浮点数    %c 单个字符    %s字符串    %p 指针的值

## 四,什么时候用?

当你想通过外部输入数据赋值到程序变量的时候.

## 五,有什么特点?

1.可以单个或批量输入,但两个%%之间的分割内容必须按原样输入.

2.格式化字符串不能用\n作换行.直接回车就可以表示结束并自动换行.

## 思考练习

1. scanf函数和printf函数的区别是什么,那共同点呢?
2. 变量在内存中占用的空间都是一样的吗?
3. int a=1与int b=100; 请问a和b哪个占用的内存空间大?
4. 请问int在内存中占几个字节?
5. 下面打印a的值是多少?

```
#include <stdio.h>
int main(){
    int a=0;
    {
        int a=2;
        {
            a=3;
        }
    }
    printf("a的值是: %d\n",a);
}
```

2. 写一个整数乘法计算器,在控制台上输入任意两个整数会计算得出二者的乘积.





# C语言的基本运算

1. 算术运算符

(+ - \* / %)

2. 关系运算符

(> < == >= <= !=)

3. 逻辑运算符

(! && ||)

4. 位运算符

(<< >> ~ | ^ &)

5. 赋值运算符

(= 及其扩展赋值运算符)

6. 条件运算符

(? :)

7. 逗号运算符

(,)

8. 指针运算符

(\* 和 &)

9. 求字节数运算符

(sizeof)

10. 强制类型转换运算符

((类型))

11. 分量运算符

(. →)

12. 下标运算符

[ ]

13. 其他

(如函数调用运算符())

# 算术运算

一,是什么?

“ + ”加法运算符 “ - ”减法运算符 “ \* ” 乘法运算符; “ / ”除法运算符  
“ % ” 取余运算符.

二,有什么用?

数据运算

三,怎么用?

直接在两个数之间加符号就可以了.如:1+1, 2-1, 3\*1, 4/2, 5%2

四,什么时候用?

当你想进行数据运算时候.

五,有什么特点?

- 1.不同数据类型精度不一样.double>float>int. 取余运算符只能整数.+1:正数 -1:负数
- 2.若用高精度向低精度变量赋值.会丢失精度.如:int a=1.8;那么a值是1.(非四舍五入)
- 3.运算优先级是.从左到右 但: \* = / = % > + = - 如果要人为改变则加().
- 4.运算时两个数据精度不一样,会自动类型转换,且是低精度向高精度转换.
- 5.同精度运算只能得到同精度的结果.如果需要提高精度,需要强制类型转换.

# 赋值及自增自减运算

一,是什么?

“=”赋值运算符 “+=”加等于运算符 “-=”减等于运算符 “++” 自增运算符  
“--”自减运算符

二,有什么用?

改变数据的值

三,怎么用?

```
int a=1; a+=2; a-=1; a++; a--; ++a; --a;
```

四,什么时候用?

当你想改变数据的值.

五,有什么特点?

1.变量才能自我变化,如自增或自减,但常量不能.

2. ++a,与a++,对a本身没有区别,但是参与其它运算时,a++这个整体的值没有变化.

如:int a=1; int b=a++;那么a=2,b等于1. 但如果是int b=++a;则a和b都为2.即++在前,则运算前自增,再参与别的运算. 但++在后.则先参与别的运算后再自增.

# sizeof

一,是什么?

sizeof是C语言的一个关键字,是一种单目操作符.而不是函数

二,有什么用?

sizeof操作符以字节形式给出了其操作数或数据类型的存储大小

三,怎么用?

sizeof(类型说明符, 数组名或表达式) 如sizeof(int);

或sizeof 变量名 如: int a;sizeof a;

四,什么时候用?

当需要查看数据类型或变量在当前编译环境占用字节数时.

五,有什么特点?

1.sizeof是根据当前系统的编译环境去计算字节的大小.

2.注意sizeof后面直接连字符的用法.会自动变成ASCII对应的int值

# 关系运算

一,是什么?

“ == ” “ > ” “ < ” “ >= ” “ <= ” “ != ”

二,有什么用?

对数据进行比较,看是否成立.

三,怎么用?

把运算符放在两个要比较的数据中间就可以.条件成立=真(1),条件不成立=假(0).

四,什么时候用?

当需要对两个数据进行比较.

五,有什么特点?

1.关系运算符运算结果是一个int值.0或1.

2.优先级:>,<,>=,<=,优先级相等, 但大于==和!=.同时==与!=优先级相等.

3.如果优先级相同的两个运算,先后顺序看结合律(结合律指要么从左往右运算,要么从右往左运算),关系运算符结合律是从左往右,即:先看优先级,再看结合律.如下:

int b=1!=2>3<4; 等效于:2>3为假=0. 0<4为真=1, 1!=1为假=0.所以最终结果=0;

# 逻辑运算

一,是什么?

“ || ” 或 “ && ” 与(且) “ ! ” 非

二,有什么用?

对多个表达式进行并列真假判断.

三,怎么用?

|| 只要有一边的表达式为真,则结果为真1,两边都不成立,则结果为假0.而&&必须两边全部为真才为真,只要有一边的表达式为假,结果为假.如: $1>2||2>1$ ;结果为真(1),但如果 $1>2&&2>1$ .结果为假(0);!则是取当前结果的反面.真假调换.

四,什么时候用?

当需要对对多个表达式进行并列真假判断.

五,有什么特点?

- 1.所有数据都有真假值.0为假,其它的数都为真(规定). 结合律:从左往右执行,
2. 如果||表达式左边表达式已经为真,则右边不执行.而&&只要左边为假右边不执行.

3.优先级顺序为: 小括号() $>$  负号(-) $>$  ! $>$  算术运算符 $>$  关系运算符 $>$  && $>$  ||

如: $1+2<3||!-3<4&&5>0$  等效于  $(1+2)<3 || ( (!(-3) )<4 \&\& (5>0) )$  结果是:1.



# 操作/运算符的优先级与结合律

优先级	运算符	结合律
从 高 到 低 排 列	() [] -> .	从左至右
	! ~ ++ -- (类型) sizeof	从右至左
	+ - * &	从左至右
	* / %	
	+ -	从左至右
	<< >>	从左至右
	< <= > >=	从左至右
	== !=	从左至右
	&	从左至右
	^	从左至右
		从左至右
	&&	从左至右
		从右至左
	?:	从右至左
	= += -= *= /= %= &= ^=  = <<= >>=	从左至右

# 思考练习

1. `int a=1.1;` 请问 `a` 打印的结果是什么？
2. `float a=1;` 请问 `a` 打印的结果是什么？
3. `double a=1/3;` 请问 `a` 打印的结果是什么？
4. `int a=3++;` 这个正确吗？
5. `int a=1; int b=a++; int c=++a;` 请问 `a b c` 三个的值是多少？
6. `int a=3!=2>1;` 请问 `a` 的值是多少？
7. `int a=0; int b=1>2||a++;` 请问 `a` 和 `b` 的值是多少？ `int b=1>2||++a` 呢？ `int b=1<2||++a` 呢？
8. `int a=!-4<0||3&&2>5;` 请问 `a` 的值是多少？
9. `int a=3?2>3:5+5;` 请问 `a` 的值是多少？
10. `int a='a';` `a` 的结果是多少？ `sizeof 'a' == sizeof(char)`, 对还是错？
11. 警告(warning) 与 错误(error) 的有什么区别？





黑马程序员  
itheima.com

——为莘莘学子改变命运而讲课  
为千万人少走弯路而著书！

iOS

# 流程控制语句

难易程度:★★★★

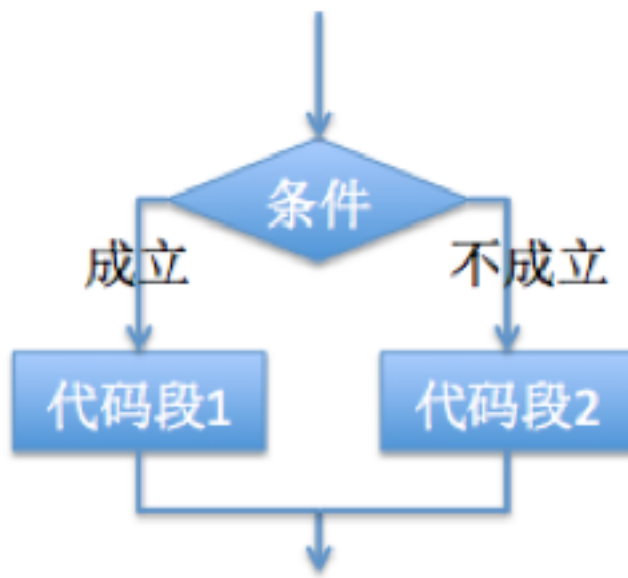
重要程度:★★★★★



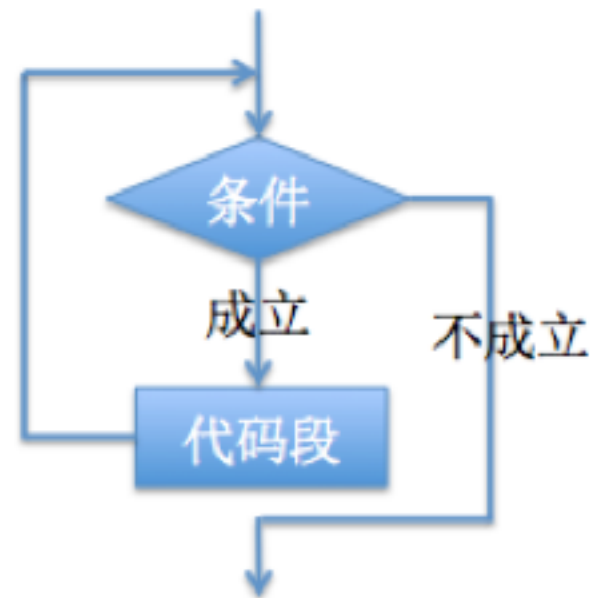
# C语言的流程控制



顺序结构



选择结构



循环结构

# 选择结构之if语句

## 一,是什么?

是指编程中用来判定所给定的条件是否满足, 根据判定的结果(真或假) 决定执行给出的两种操作其中的一个

## 二,有什么用?

可以根据条件执行不同的操作.

## 三,怎么用?

1.if(){}    2.if(){}else{}    3. if(){}else if(){}    4.if(){}else if(){}else{}    5.if() ;

## 四,什么时候用?

需要根据条件执行不同的操作

## 五,有什么特点?

- 1.一次if语句只能执行给出的所有操作中的一个.
- 2.if();这种用法时if语句只有紧跟后面一句有效.(读到下一个;号结束).且不能定义变量
3. 除了if();这种用法,所有的if()与{}之间不能有;号
- 4.if(条件判断){},里面的条件判断如果是与常量的等于判断,则常量写在前面(建议);

# if语句练习

```
if (4) {  
    printf("A\n");  
}
```

```
int a=0;  
if (a=0) {  
    printf("A\n");  
}
```

```
int a=0;  
if (a==0)  
    int b=0;  
printf("A\n");
```

```
int a=0;  
if (a=0)  
    printf("A\n");  
printf("B\n");
```

```
int a=0;  
if (a++){  
    printf("A\n");  
}else{  
    printf("a=%d\n",a);  
}
```

```
int a=0;  
if (a++){  
    printf("A\n");  
}else if(a++){  
    printf("a=%d\n",a);  
}
```

```
int a=0;  
if (a++){  
    printf("A\n");  
}else if(a++){  
    printf("a=%d\n",a);  
}else if(a>1){  
    printf("a=%d\n",a);  
}
```

编程题:

1.任意输入一个整数成绩,如果大于等于80在控制台上打印优秀,大于等于70小于80分打印良好.大于等于60分小于70分打印及格,小于60分打印不及格.

2.在屏幕上提示用户输入数字 ,输入1输出打印你好! 输入2 打印hello!



# 选择结构之switch语句

## 一,是什么?

是指编程中用来判定所给定的条件是否满足, 根据判定的结果(真或假) 决定执行给出的具体操作

## 二,有什么用?

可以根据不同的int值匹配执行不同的操作.

## 三,怎么用?

switch(表达式){ case 常量表达式1: 语句1;break;... case...;default: 语句n;}

## 四,什么时候用?

当是判断int值匹配不同的值执行对应的操作时可以用, 简化代码结构.

## 五,有什么特点?

- 1.条件结果只能是整数,而case只能是整型常量.注意:case 'a';
- 2.break 可有可无,有则执行完跳出,没有则继续执行,直到遇到下一个break或执行完.
- 3.default可有可无,有则如果条件都不满足,会执行这句, 无则条件都不满足,执行结束.
- 4.case下面要定义变量必须加{}.且case后的常量值不能重复
- 5.switch语句可以全部用if语句替代.但if语句不能全用switch替代.



# switch语句练习

下面写法是否正确,如果正确打印结果是什么？

```
switch (int a=2) {  
    case 1:  
        printf("A\n");  
        break;  
    case 2:  
        int b=0;  
        printf("B\n");  
        break;  
    case 3.2|:  
        printf("C\n");  
        break;  
}
```

```
int a=0;  
switch (a=2) {  
    case 1:  
        printf("A\n");  
    case 2:  
        printf("B\n");  
    case 3:  
        printf("C\n");  
        break;  
}
```

```
int a=0;  
switch (++a) {  
    case 1:  
        a++;  
        printf("A\n");  
        break;  
    case 2:  
        printf("B\n");  
        break;  
}
```

编程题:

- 1.任意输入一个整数成绩,如果大于等于80在控制台上打印优秀,大于等于70小于80分打印良好.大于等于60分小于70分打印及格,小于60分打印不及格.
- 2.在屏幕上提示用户输入数字 ,输入1输出打印你好! 输入2 打印hello!



# 循环结构之while与do while语句

一,是什么?

是指编程循环语句, 当满足条件时进入循环, 循环判断,直到不满足条件跳出循环。

二,有什么用?

可以根据条件反复执行一段代码

三,怎么用?

`while(<条件>) {<语句>;    do {<语句>} while(<条件>);`

四,什么时候用?

当需要重复执行一段代码或满足一定条件重复执行的一短代码实现相应的功能.

五,有什么特点?

1.只有满足条件循环才会执行.

2.注意死循环的出现.会一直循环执行下去,除非程序强制退出.

3.如果碰到**continue**,则跳出本次循环继续下一次判断,如果条件成立继续执行循环.

4如果碰到**break**无论条件满足与否,循环立刻结束.

5.**continue**与**break**跳出的都是当前所在的循环作用域.如果外层有循环,不受影响.

6.**do while**至少执行一次.而**while**不一定.

# while语句思考练习

- 1.什么情况下会出现死循环?终端下死循环怎么退出?
- 2.循环中**break**与**continue**的作用分别是什么?
- 3.计算 $1+2+3+....+100$ 的值.
- 4.计算出100以内所有的奇数的个数.
- 5.以下循环的次数是多少?

```
int i=0;
while(i<10){
    if(i<1)
        continue;
    if(i==5)
        break;
    i++;
}
```



# 循环结构之for语句

## 一,是什么?

是指编程循环语句,当满足条件时开始循环判断执行,直到不满足条件跳出循环.

## 二,有什么用?

可以根据条件反复执行一段代码

## 三,怎么用?

`for(;;){}`里面是3个语句,两个分号。第一个语句是开始前执行,第二个语句是判断真假,如果真,就执行后面(大括号内)的代码。第三个语句是每次执行完毕后执行

## 四,什么时候用?

当需要重复执行一段代码或满足一定条件重复执行的一短代码实现相应的功能.

## 五,有什么特点?

- 1.只有满足条件循环才会执行.
- 2.注意死循环的出现.会一直循环执行下去,除非程序强制退出.
- 3.如果碰到**continue**,则跳出本次循环继续下一次判断,如果条件成立继续执行循环.
- 4如果碰到**break**无论条件满足与否,循环立刻结束.
- 5.**continue**与**break**跳出的都是当前所在的循环作用域.如果外层有循环,不受影响.



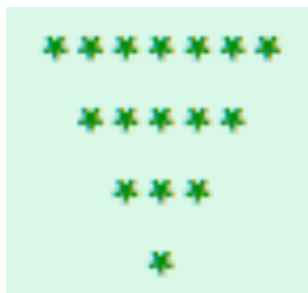
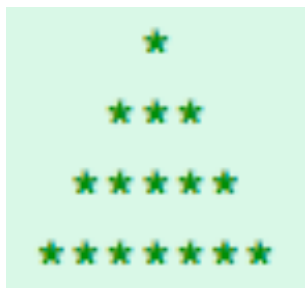
# for语句思考练习

1. 请问break与continue有什么区别？
2. for语句与while语句可以互换吗？二者的相同点及区别是什么？
3. 以下两种写法正确吗？

```
for (int i=0; i<10; i+=2)
    int a=2;
```

```
for (int i=0; i<10; i+=2,a++){
    int a=2;
}
```

4. 计算1+2+3+....+100的值(用for语句实现).
5. 打印一个九九乘法口诀表.
6. 打印下面两个图形.





黑马程序员  
itheima.com

——为莘莘学子改变命运而讲课  
为千万人少走弯路而著书！

iOS

# 函数

难易程度:★★★

重要程度:★★★★★

# C语言函数

## 一,是什么?

是指编程中对一段功能代码的封装,可以理解为一个程序片段.

## 二,有什么用?

可以封装代码,方便调用,提高复用性

## 三,怎么用?

定义:返回值类型 函数名(形式参数类型 参数名1,.....){ 具体代码 return 返回值}

调用:函数名(实际参数1,实际参数2,.....);

## 四,什么时候用?

当需要去封装一段代码实现一个对应的功能时候.

## 五,有什么特点?

- 1.返回值类型不定义默认是int.如果不需要具体返回值类型用void(return可以省略).
- 2.是否要定义参数看是否有未知内容参与运算,调用时参数必须对应.参数传递的是值.
- 3.return可以有多个,但是如果执行到return就不会再执行后面的代码,函数执行结束.
- 4.函数不调用不执行,执行顺序看调用顺序.定义了返回值类型可以不返回.(不建议).
- 5.函数必须定义才能调用,但不能重复定义.不能嵌套定义.不能自己调用自己(死递归).

# 函数思考练习

- 1.函数返回值一定要有吗?有返回值类型一定要有return吗?
- 2.函数结束的标志是什么?没有返回值类型可以不不要return?
- 3.是否需要返回值看什么?返回值怎么取?
- 4.函数可以重复定义吗?参数不一样可以吗?
- 5.函数可以自己调用自己吗?有没有例外?
- 6.函数不定义可以用吗?可不可以编译成.o文件?
- 7.形式参数与实际参数有什么区别?
- 8.是否需要参数看什么?
- 9.函数调用时传递参数实际传递的是什么?
- 10.以下写法是否确?

```
void hello(int a){  
    int a=0;  
    printf("%d\n",a);  
}
```

- 11.实现一个加法计算器,可以反复使用,当两个要相加的数都为0时,程序退出.



黑马程序员  
itheima.com

——为莘莘学子改变命运而讲课  
为千万人少走弯路而著书！

iOS

# 进制及内存分析

难易程度:★★★★

重要程度:★★★★



# 编程中进制的用法

## 1. 代码表示:

二进制: **0b** 或者 **0B** 开头 八进制: **0** 开头 十进制: 直接写 十六进制: **0x** 或 **0X** 开头 (最大 **F**)

## 2. 进制的打印输出:

**%o**: 表示输出不带符号八进制整数

**%x**: 表示输出不带符号十六进制整数

**%d**: 表示输出有符号十进制整数

**%u**: 表示输出不带符号十进制整数

注意: 二进制不可以直接输出. 有符号与无符号区别: 有符号可以有负数表现 (如: **-2**)

## 3. 进制数的在内存中的表现形式:

都是二进制. 一个字节代表一个 **8** 位的二进制. 占用 **X** 个字节, 就表示占用 **8\*X** 位二进制

## 4. 判断下列数据正确与否?: 注意: 1. 先确定是几进制. 2. **e3** 表示 **10** 的 **3** 次方

00110	0x0011	0b110	0x7H4	110	0986	0x6G9	0b102	.089	-109
+188	0xFadA	16f	16.0f	16.0F	-.016	1.6e3	16e3.3	1.6e-3	.011

# 位运算

一,是什么?

是直接对整数在内存中的二进制位进行操作的方法(**只能操作整形数据**)

二,有什么用?

直接通过操作二进制的位来实现运算,高效

三,怎么用?

**& 与运算:**对应两个二进位均为1时,结果位才为1,否则为0。(有假为假==0)

**| 或运算:**对应的两个二进位有一个为1时,结果位就为1,否则为0。(有真为真==1)

**^ 异或运算:**对应的二进位不同时为1,否则为0。(不同为真==1,相同为假==0)

**注意:**1.相同整数相^的结果是0.2.一个数^另一数两次结果不变3.整数相^跟顺序无关

**~ 取反运算:**对整数的各二进位进行取反,包括符号位(0变1, 1变0,真假互换)

**<< 左移:**把整数的各二进位全部左移n位,高位丢弃(**包括符号位**),低位补0。

左移n位其实就是乘以2的n次方(但移动后符号位变化除外,符号位改变则正负改变)

**>> 右移:**把整数的各二进位全部右移n位,**符号位不变**。高位的空缺是正数补0;

是负数高位是补0或是补1 取决于编译系统的规定;右移n位其实就是除以2的n次方





# 进制及内存分析思考练习

1. 一个字节有多少位二进制？如果一个数据占两个字节那么有几位二进制？
2. `int a=1;`如果是64位下.在内存中创建过程是怎么样？怎么表示？
3. 二进制,八进制,十进制,十六进制怎么区分？
4. 进制转换中几位二进制能表示一个八进制？十六进制呢？
5. 怎么确定一个数据类型的取值范围？
6. 类型说明符中各类型能互相修饰吗？
7. 位运算能操作浮点型数据吗？
8. `&, |, ^, ~, <<, >>`.分别叫什么？有什么特点？
9. `printf("\\");`这样写对吗？这样呢 `printf("%c\\n", '\\');`
10. `int a='a'+1; char c='a'+1;`请问a和c的打印结果是什么？可不可以用%c输出int数？

## 编程题

1. 用位运算实现两个整数值的互换.
2. 用位运算判断一个整数的奇偶性.
3. 编写一个函数输出一个整数的二进制表现形式.



黑马程序员  
itheima.com

——为莘莘学子改变命运而讲课  
为千万人少走弯路而著书！

iOS

# 数组

难易程度:★★★★

重要程度:★★★★★

# 数组

## 一,是什么?

是用来存储一组数据的容器.

## 二,有什么用?

用于把同一类型的一组数据统一管理起来.

## 三,怎么用?

**定义:** 数据类型 数组名[长度(数量)]; **初始化:** 类型 数组名[(长度)]={元素1,元素2,...}

**赋值:** 数组名[角标(索引)]=值; **使用:** 数组名[角标]; 如 `int a=arr[0];`

## 四,什么时候用?

当需要管理一组**有**关联****的数据时.

## 五,有什么特点?

- 1.数组必须要定义长度(正整数),定义时可初始化(长度必须是常量)或之后单独初始化.
- 2.数组是有序的.每个数据都有角标.从0开始分配.角标**不能**`>=`数组长度.类型一致.
- 3.数组有地址(数组名),里面的元素也有地址.数组当参数时(长度可省)传递的是地址.
- 4.数组当参数传递时是指针类型.所以如果需要用数组长度要额外传递.

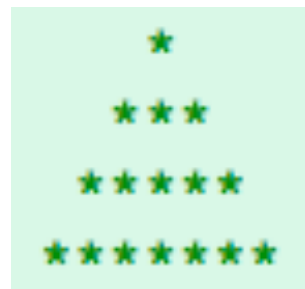
# 数组思考练习

- 1.数组可以存不同类型的数据吗？
- 2.数组定义时长度可以不确定吗？
- 3.数组一定要全部初始化后才能使用吗？
- 4.数组一般什么时候用？
- 5.数组的角标从几开始,角标最大值是几？
- 6.数组的长度可以是小数吗？
- 7.数组有没有地址?里面的元素呢？
- 8.数组可不可以有重复元素？
- 9.数组当参数传递时传递的是什么？
- 10.数组当参数传递时能获取长度吗？

编程题:

已知有一个int数组:`int arr[6]={1,2,4,3,8,5};`

- 1.编写一个函数求这个数组的最大值:
- 2.给这个数组从大到小排序.(两种方式)
- 3.用一个二维数组实现打印右边的星形.





黑马程序员  
itheima.com

——为莘莘学子改变命运而讲课  
为千万人少走弯路而著书！

iOS

# 字符串

难易程度:★★★

重要程度:★★★★★

# 字符串

一,是什么?

是一串字符,也是一个以'\0'结尾的字符数组.

二,有什么用?

用于表示一串字符或文本数据.

三,怎么用?

**定义及初始化:** `char arr[]="abc";`或`char arr[4]={'a','b','c','\0'}`

**输出:**`printf("%s\n",s);`或`printf("%s\n",&arr[0]);` **赋值:**`strcpy(字符变量名,"字符串");`

四,什么时候用?

当需要表示一串字符或文本的时候.

五,有什么特点?

- 1.后面必须有'\0'结尾.否则只算普通的字符数组.但'\0'不会输出,只表示字符串结束.
- 2.字符串输出占位用%s必须遇到\0.才能结束,否则会连续输出更高位地址值的字符.
- 3.strlen函数用于计算一个字符串的长度(字符数量),使用必须引入<string.h>.
4. strlen不会计算\0. 且碰到\0结束,但是sizeof不受\0影响,且长度会包含\0.
- 5.字符串一定是字符数组,但字符数组不一定是字符串.

# 字符串思考练习

- 1.字符串结束的标识是什么？
- 2.怎么输出`char arr[]="abc";`中的'a'？
- 3.`char arr[]={ 'a','b' };`与 `char arr[]="ab"`的区别？
4. `char arr[]="ab\0c";strlen(arr)`的值是多少？
- 5.`sizeof`与`strlen`的区别？
- 6.字符串的地址跟自己的哪一个字符地址一致？

编程题：

- 1.定义一个字符串数组.并打印所有字符串.
- 2.编写一个函数判断一个字符串中是否包含某个字符,包含返回1,不包含返回0.  
(用两种方法).



黑马程序员  
itheima.com

——为莘莘学子改变命运而讲课  
为千万人少走弯路而著书！

iOS

# 指针

难易程度:★★★★★

重要程度:★★★★★



# 指针

## 一,是什么?

严格上来讲,指针具有两层含义:一是指一种数据类型.二是指一种保存地址的变量.

## 二,有什么用?

用于存储某一个地址,从而可以通过地址直接对该地址内存空间中的数据进行操作.

## 三,怎么用?

**定义:** 类型 \*指针名;如: `int *p;`表示定义一个能存int类型地址的指针变量, 名字叫p.

**赋值:** 指针变量名=指针(地址).注意:指针获取方法(&变量名).

## 四,什么时候用?

当需要存储地址,以便于操作该地址及地址对应的内存数据的时候.

## 五,有什么特点?(注:以下的p表示一个指针变量名)

- 1.指针**本身**也有地址.指针定义之后,自身地址不变,**但里面的存的地址**可以变化.
2. **&p**表示指针自身地址,而**p**表示指针存的地址 **\*p**表示指针所存地址的内存空间.
- 3.指针不初始化赋值不要使用.未赋值前地址是随机的.未赋值不可以用**\*p**.
- 4.**\*修饰什么,就表示是什么的指针.判断原则:先确定变量的类型.(根据符号优先级)**
- 5.指针是什么类型的,就应该存什么类型的地址.否则**\*p**取对应内存空间的值会出错.

# 指针思考练习

1. 指针里存的实际是什么？
2. 指针本身有地址吗？
3. 指针占几个字节？
4. 如何取指针里的数据？
5. 一个类型的指针可以存不同类型的地址吗？
6. 指针可以直接操控指向空间的数据吗？
7. 如何表示一个指针的指针？

编程题：

1. 实现一个函数能够实现两个整数的加减乘除。
2. 已知 `int a=5; int *p=&a;` 请输出指针本身的地址, 及指针的值, 及其所指内存空间的值。

# 指针与数组

## 一,什么是指针数组?

表示一个数组,用来存指针的数组就叫指针数组.数组里面的元素都是指针.

如:`int *p[3];`表示定义一个可以存三个int指针的数组.

赋值: `int a1=1,a2=2,a3=3; int *p[3]={&a1,&a2,&a3};` 取值: `int *a=p[0];`

## 二. 什么是数组的指针?

表示一个指针,而这个指针类型是数组.

如:`int (*p)[3];`表示这是一个含有三个元素的int数组的指针.

赋值:`int arr[3]={1,2,3}; int (*p)[3]=&arr;`

## 三.什么是数组元素的指针?

表示一个指针,而这个指针存的是数组一个元素的地址. `int arr[1]={1}; int *p=&arr[0];`

## 三,有什么特点?

- 1.数组名表示第一个元素的地址,&数组名表示整个数组的地址.二者值相同,但意义不同.
- 2.指针可以运算.但仅限于加减运算.运算的实质就是地址运算.即地址的移动.
- 3.地址(+1)向高位移动,而(-1)向低位移动.不同指针类型移动一位(+1或-1)的字节数不同.
- 4.实际移动的字节数看指针类型,类型占多少个字节,一位就移动多少个字节.

# 指针与二维数组

## 一,什么是二维指针数组?

是一个数组.里面的每一个子数组存的元素的类型是指针.

例如: `int a=0,b=1,c=2,d=3,e=4,f=5;` //定义int型变量  
//获取所有变量的指针

```
int *pa=&a;    int *pb=&b;    int *pc=&c;
int *pd=&d;    int *pe=&e;    int *pf= &f;
```

//定义这个二维指针数组.

```
int *p[2][3]={{pa,pb,pc},{pd,pe,pf}};
```

## 一,什么是二维数组的指针?

是一个指针,只是指针类型是二维数组.

例如: `int arr[2][3]={{1,2,3},{4,5,6}};` //这是一个二维数组  
`int (*p)[2][3]=&arr;` 就是一个二维数组的指针.



# 指针与数组思考练习

1. 可以把数组叫成指针吗？二者一样吗？
2. 说说指针数组与数组指针的区别？分别怎么表示？为什么？
3. 指针可以运算吗？能进行什么运算？
4. 指针运算+1移动的字节数怎么确定？
5. 数组当参数运算实际传递的是什么？
6. 数组的指针与数组第一个元素的指针有什么关系？二者是同一类型吗？
7. 如何确定一个变量的类型？

编程题.

1. 定义一个指针数组,并把它值赋值给一个指针.
2. 用指针实现数组(`int arr[3]={1,2,3}`)所有元素的打印.

经典面试题.

下面打印出来的结果是什么？

```
int a[] = {1, 2, 3, 4, 5};  
int *p = (int*)&a + 1;  
printf("%d, %d\n", *(a+1), *(p-1));
```

# 指针与字符串

一,怎么表示一个字符串?

```
char arr[] = "heima";
```

二,怎么用指针表示一个字符串?

```
char *p = "heima";
```

三,怎么表示一个字符串数组?

```
char arrs[2][8] = {"itcast", "heima"};
```

四.怎么用指针表示一个字符串数组?

```
char *arrs[2] = {"itcast", "heima"};
```

五,有什么特点?

- 1.指针定义的字符串是常量,不能被改变.而字符数组定义的是变量.可以被改变.
- 2.指针定义一个字符串,指针存的实际是在常量区的那个字符串的首字符的地址.



# 指针与字符串思考练习

1. 指针可以定义字符串吗？
2. 指针定义字符串与字符数组定义的区别是什么？
3. 指针定义字符串那么,这个指针中存的是什么？
4. 什么时候需要用指针表示字符串？
5. 指针型字符串与普通字符串常量的区别是什么？
6. `char *str="123abc";`里的字符串能改成123def吗？

编程题:

1. `char *str="123abc";`请用指针遍历这个字符串的每一个字符.
2. 把 `char *str="123abc";` 变成一个字符数组.

# 指针与函数

## 一,什么是指针型函数?

是函数. 这个函数的返回值类型是指针;

**定义:**类型 \* 函数名(参数){return 指针类型;}

如: `int *add(int *a){return &a;}`. 表示定义一个返回值是int指针类型的add函数.

## 二,什么是函数指针?

是指针. 函数也会在内存开辟一块空间. 函数指针即这块空间的地址.

**定义:**函数返回值类型 (\*指针名)(参数类型); 赋值: 指针名=函数名;

如: 先定义函数 `int add(int a){ //... }. int (*p)(int)=add;`

**使用:** 指针名(实参); 如: `p(3);` 等效于 `add(3);`

## 三,有什么特点?

1. 函数指针做加减运算可以, 但没有意义. 不建议.

2. 函数指针可以当作参数传递. 因为其是一个指针类型.

3. 不管什么类型的指针, 只要是指针, 其里面存的就是地址.





# 指针与函数思考练习

1. 什么时候需要定义一个指针型函数？
2. 什么时候需要定义一个函数的指针？
3. 执行函数时,会在内存开辟空间吗？
4. 一个程序里的一个变量的指针在程序每次运行时结果都一样吗？

编程题.

1. 定义一个指针型函数,并调用.
2. 定义一个函数的指针,并使用.

# 全局变量

## 一,是什么?

在所有函数外部定义的变量,称为全局变量。.

## 二,有什么用?

用于存储一个需要长期存于内存,并需要改变的变量.

## 三,怎么用?

跟普通变量没区别,只在位置在所有函数之外.

## 四,什么时候用?

当需要定义一个随程序一直存于内存中,并可以被其它函数共享的变量时候.

## 五,有什么特点?

- 1.生命生期跟随程序.程序启动创建,程序退出其才会销毁
- 2.作用域是其所在位置之下的所有函数,都可以共享并改变.
- 3.变量的使用遵循就近原则如果函数内有同名变量,那么用函数自己的.
- 4.如果没有初始化赋值,默认值是0;



# 全局变量思考练习

- 1.全局变量的生命周期是怎样的？
- 2.全局变量的定义的位置在哪里？
- 3.全局变量的作用域是怎么样？
- 4.如果函数内有与全局变量同名变量用哪个？
- 5.全局变量与局部变量区别是什么？
- 6.全局变量有什么应用场景?(举个例子)

编程题:

实现一个函数调用本身不会出现死递归的案例.



黑马程序员  
itheima.com

——为莘莘学子改变命运而讲课  
为千万人少走弯路而著书！

iOS

# 结构体

难易程度:★★★★

重要程度:★★★★

# 结构体

## 一,是什么?

是一种数据类型.是构造类型.里面可以存不同类型的数据.

## 二,有什么用?

用于封装不同的数据类型到一个结构中.方便使用.

## 三,怎么用?

**定义:**1.定义结构体类型 `struct 结构体类型{数据类型1 名称;数据类型2 名称;...};`

2.定义结构体变量及赋值: `struct 结构体类型 变量名={数据1,数据2....};`

**取值:**变量名.结构体元素变量名;      **赋值:**变量名.结构体元素变量名=新值;

## 四,什么时候用?

当需要把相关联的不同数据类型封装起来一起使用的时候.

## 五,有什么特点?

- 1.可在定义结构体类型时同时定义结构体变量名,可定义匿名类型结构体;
2. 同一作用域内不可重复定义结构体类型和结构变量. 匿名类型结构体除外;
- 3.可在函数内也可在函数外,效果及用法类似于局部变量和全局变量. 也可嵌套定义.
- 4.可同时在函数内外定义同一个类型,使用同样遵循就近原则,结构体变量也是如此.

# 结构体数组

## 一,是什么?

是数组,只不过是数组里每一个元素是结构体类型。

## 二,有什么用?

用于存储多个有关联的结构体。

## 三,怎么用?

定义及初始化赋值: `struct 结构体类型 数组名[长度]={结构体1,结构体2};`

如: `struct Person{char *name;int age;};`  
`struct Person per={"itheima",2};      struct Person per1={"itcast",3};`  
`struct Person arr[2]={per,per1};`

取值: `数组名[角标];` 单独赋值: `数组名[角标]=新的结构体;`

如: `struct Person per3=arr[0];`

## 四,什么时候用?

当需要存储多个有关联的结构体,并作为一个整体使用时。

## 五,有什么特点?

### 1.遵循数组的特点

# 结构体的指针

## 一,是什么?

是指针,只不过是结构体类型。

## 二,有什么用?

用于存储结构体的地址,并操作地址对应的结构体空间里的数据。

## 三,怎么用?

定义及初始化赋值: `struct 结构体类型 *指针名=&结构体变量`

如: `struct Person{char *name;int age;}; struct Person per={"itheima",2};`  
`struct Person *p=&per;`

取值: `(*指针名).结构体元素名;` `指针名->结构体元素名;` (结构体独有的用法)

赋值: `(*指针名).结构体元素名=新值;` `指针名->结构体元素名=新值`

如: `(*p).name,(*p).age;`或`p->name,p->age;`

## 四,什么时候用?

当需要使用指针操作结构体时。

## 五,有什么特点?

1.指针移动对结构体本身没有意义.结构体名并不包含第一个元素的地址。

# 结构体思考练习

1. 结构体是什么数据类型？
2. 结构体里面的元素可以是不同类型吗？
3. 结构体可以定义在什么位置？使用时遵循什么原则？
4. 结构体类型可不可以重复定义？有没有例外？
5. 结构体在函数内和在函数外生命周期一样吗？
6. 结构体与数组有什么区别？
7. 结构体数组是什么？
8. 结构体指针取值有哪两种方式？

编程题：

1. 定义一个结构体，然后打印里面的每一个元素。（两种方式：结构体变量和指针）
2. 用结构体定义一个三口之家。家里每个成员都有：姓名，年龄，身份（如：父亲）。
3. 打印这个三口之家的所有成员信息。





黑马程序员  
itheima.com

——为莘莘学子改变命运而讲课  
为千万人少走弯路而著书！

iOS

# 枚举

难易程度:★★

重要程度:★★

# 枚举

## 一,是什么?

是一种数据类型.称为枚举类型.是构造类型的一种,里面存的都是有别名的整型常量.

## 二,有什么用?

可以把多个整型常量以别名的形式封装成一个整体,便于别的变量识别并取值使用.

## 三,怎么用?

**定义:**1.定义枚举类型 `enum 枚举类型{元素1, 元素1, ...};` 或 `{元素1=0,元素2=1,}`

2.定义枚举变量及赋值: `enum 枚举类型 变量名=对应枚举里的一个元素名;`

**取值:**变量名=枚举元素名;

## 四,什么时候用?

当一个变量只有固定的多个int取值范围时.

## 五,有什么特点?

1.枚举类型只能是int值,且不能直接用int常量,需要用别名(建议全部大写)

2.如果不设定值,那么默认第一个元素是0,以此类推;

3.同一个作用域内不能出现重复的元素.不同的枚举类型也不行.

# 枚举思考练习

- 1.枚举的元素是什么类型?可以有多种类型吗?
- 2.枚举元素需要定义类型吗?
- 3.枚举里的元素可不可以不赋值直接使用?
- 4.枚举类型同一作用域内可以有重复元素吗?
- 5.枚举里的元素定义好后,对应的值可以改变吗?
- 6.枚举能用循环去遍历吗?

编程题:

实现一个定义描述星期的枚举类型.并取周二的值.



黑马程序员  
itheima.com

——为莘莘学子改变命运而讲课  
为千万人少走弯路而著书！

iOS

# 预处理指令

难易程度:★★★

重要程度:★★★



# 预处理指令教学大纲与目标

一. 文件包含 掌握

二. 宏定义 熟悉

三. 条件编译 了解

# 宏定义

一,是什么?

是一个替换代码的预处理指令

二,有什么用?

可以在编译之前进行代码替换.

三,怎么用?

开始替换: **#define** 被替换内容 替换成的内容

如: **#define kLength 6** 表示从这行开始下面所有代码中的kLength 替换成6

结束替换: **#undef** 被替换内容 如: **#undef kLength** 表示到一行, 宏定义失效

四,什么时候用?

当需要用一些重复性代码或简单的函数运算时,可以用宏定义.提高效率

五,有什么特点?

1.命名尽量以大写或小写k开头.便于区分.(规范)

2.被双引号的引起来的不会被替换

3.宏定义只会做代码替换,不会关注任何逻辑,容易出现优先级错误(加括号解决).

4.替换可以设置作用域.如果不设置,作用域则是这行代码出现到代码的最后一行.



# 预处理指令思考练习

- 1.什么是预处理指令？
- 2.C语言中的预处理指令有哪些？
- 3.预处理指令出现的意义是什么？
- 4.文件包含关键字是什么？核心作用是什么？
- 5.宏定义关键字是什么？核心作用是什么？
- 6.条件编译关键字是什么？核心作用是什么？

编程题：

- 1.用宏定义实现取两个数的最小值.
- 2.用预处理指令实现防止出现文件重复包含及交叉包含

# typedef

## 一,是什么?

是一个C语言的一个关键字.

## 二,有什么用?

可以给一个数据类型取一个别名,方便使用.

## 三,怎么用?

定义: typedef 类型 别名;

如: typedef long int Lint; 表示给一个long int 类型取一个别名为Lint

则: long int a =5;与Lint a=5;二者等效

## 四,什么时候用?

当希望给一个类型取一个别名时(类型比较复杂等),可以用,方便理解调用.

## 五,有什么特点?

- 1.定义位置没有限制,函数内外都可以.
- 2.作用域为定义位置开始至所处作用域结束.类似于变量
- 3.可以用于任意数据类型.
- 4.别名具有数据类型的含义,这是跟宏定义最大区别.



# static与extern

## 一,是什么?

是C语言的关键字.

## 二,有什么用?

控制一个变量或函数的作用域.

## 三,怎么用?

**修饰函数:**static 返回值 函数名();表示函数为内部函数只能被本文件访问.

**extern** 返回值 函数名();表示函数为外部函数能被程序的所有文件访问.(默认)

**修饰全局变量:** static:表示变量只能被当前文件访问. **extern:**所有文件共享.(默认)

**修饰局部变量:** static:表示延长变量生命周期至程序结束. **extern:**声名一个全局变量.

## 四,有什么特点?

- 1.外部函数不同文件中也不能重名,但内部函数可以
- 2.同类型全局变量可以重复定义,但在内存中只有一份.
- 3.可以在函数内声名一个全局变量.但要使用必须在函数外定义. **extern**只是**声名**.
- 4.**static**修饰定义局部变量,则这个局部变量生命周期延长至程序结束. 但作用域不变.

# C语言总结

