C语言程序设计

**一、初阶C语言学习**

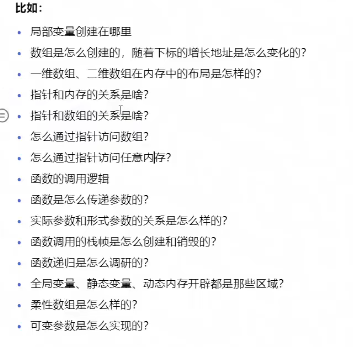
**课前准备：**

**1.**学会写博客，技术文章，技术学习的理解、讲解、心得、总结（公司看重）。CSDN，博客园。**2.**github的重要性https://github.com，国内码云<https://gitee.com>，自己学习翻墙。**3.**你是学过编程的，怎么学习的，学了什么，怎么证明，github贡献度。注册账号使用，代码仓库，QQ相册类比，代码仓库自己的账号地址。比特鹏哥github使用教程：注册+提交代码。**4.**做笔记，有道云笔记，印象笔记。思维导图-C语言初阶-知识点-难点，从大到小。**5.**遇到问题，自己尝试解决。**6.**学好编程，一系列学习，一门语言深入学习，听课看书实践。

**学好C语言：**

第一，必须实践。1.学习语法规则，不同的语言具有不同的标准和规则。2.学习语法的规则，学习书或者视频。3.动手写代码，在实践中找错。模仿阶段，模仿别人优秀写的代码，模仿不是抄写，而是理清别人代码的思路，然后由自己写。自主动手阶段，自己思考慢慢理清思路。

第二，必须要画图理解，内存布局。C语言是一门偏底层的语言，可以直接操作和访问内存，所以我们要清楚的知道我们写出的代码所对应的内存布局。

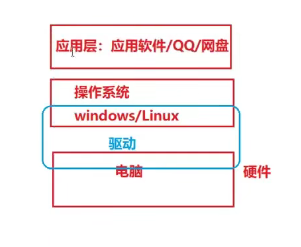


第三，调试调试调试。做到心中有数，写代码前要心中有数，实现与预期相同的效果，掌握调试技巧。初学时，调试理清逻辑，学成后，调试帮我们找出码中的错误或引起不预期的问题。（调试窗口，逐语句调试，监视窗口查看变量变化情况）

**1 初识C语言**

1.1 什么是C语言

语言是人与人之间交流，计算机语言是人和计算机交流，C语言是一种通用的计算机编程语言，广泛用于底层开发（擅长应用层往下操作系统和驱动，也可使用在上层设计）。



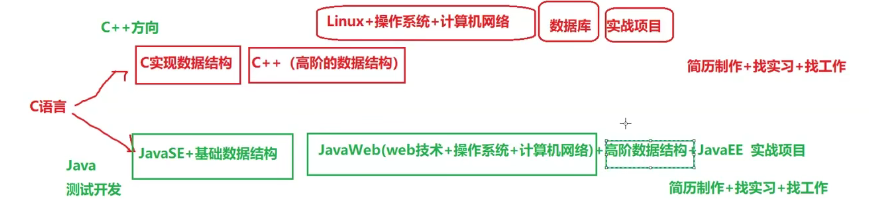
怎么写代码，编译器MSVC微软，Visual Studio2013集成开发环境，集成了VC的编写和编译器。编写第一个程序，编写代码步骤：

1.创建一个项目

2.创建一个源文件xxx.c为源文件，xxx.h为头文件

3.写代码

4.编译代码+链接+运行



1.2 数据类型

计算机语言是拿来写程序的，写程序是为了解决生活中的问题的，所以计算机编程是应该具有描述问题的能力的。比如商品价格15.6元，我们需要用浮点数描述，比如年龄20岁，我们需要用整型来描述。



#include<stdio.h>

int main()

{

char ch = 'a';//数据类型定义一个空间，存放数据

int ege = 20;

short num = 10;

long height = 170;

//long long

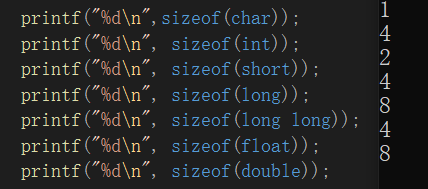
float weight = 55.5;

double b = 1.75;

return 0;

}

数据类型是用来定义变量的，对应数据类型或变量所占空间大小，可以由sizeof()关键字-操作符计算，计算单位为字节，整型数据类型的选择较多可以按空间利用较低分配。



计算机中的单位：bit 比特位，一个二进制位；byte 字节=8bit；kb、mb、gb、tb之间转换为1024.

1.3 变量与常量

1.3.1 基本概念

生活中的一些数据，有些是不可变的：血型，圆周率；有些数据是可变的：工资，年龄，体重。

C语言描述这些数据，常量与变量。类型 变量名 = 数值（推荐给个初值）；

printf()，数据输出格式%d-int整型，%f-float浮点型，%lf-double浮点型。

1.3.2 变量的分类

全局变量与局部变量，全局变量在函数外部定义，局部变量在函数内部定义，当全局变量与局部变量名字冲突时，局部变量优先。

#include<stdio.h>

int a = 100;

int main()

{

int a = 10;

printf("%d\n",a);

return 0;

}

#define \_CRT\_SECURE\_NO\_WARNINGS 1//输入函数报错设置

#include<stdio.h>

int main()

{

int a = 0;

int b = 0;

int sum = 0;

scanf("%d %d",&a,&b);//输入函数，取地址

sum = a + b;

printf("%d\n",sum);

return 0;

}

例1：编写代码找十个数之间的最大值。

#define \_CRT\_SECURE\_NO\_WARNINGS 1

#include<stdio.h>

int main()

{

int i = 0;

int max = 0;

int a[10] = {0};

printf("请输入十个整数\n");

//输入十个数

for (i = 0; i <= 9; i++)

{

scanf("%d",&a[i]);

}

//查看十个数

for (i = 0; i <= 9; i++)

{

printf("%d\n", a[i]);

}

//找十个数中的最大值

max = a[0];

for (i=0;i<=9;i++)

{

if (a[i] > max)

max = a[i];

}

printf("max=%d\n",max);

return 0;

}

例2：求十个数的平均值

#define \_CRT\_SECURE\_NO\_WARNINGS 1

#include<stdio.h>

int main()

{

float a[10] = {0};

int i = 0;

float avg = 0;

float sum = 0;

printf("请输入十个数\n");

for(i = 0; i < 10;i++ )

{

scanf("%f",&a[i]);

sum = sum + a[i];

}

printf("%f\n",a[9]);

printf("%f\n",sum);

avg = sum / 10.0;

printf("avg=%f\n",avg);

return 0;

}

面试官如何考察你：

1.笔试-技术笔试（选择＋编程）

2.面试（技术面+HR面）

技术面：问技术的理解+写代码+问项目

1.3.3 变量的作业域和生命周期

变量的作用域是变量的有效使用范围，一般包含于定义后的大括号里。由此局部变量的作业域就是定义该变量后的大括号内，全局变量的作用域是整个工程（跨文件也可使用，extern int a=100;）。

一个变量的生命周期指的是变量的创建与销毁之间的时间段，局部变量的生命周期：进入局部范围定义生命开始，到出局部范围结束；全局变量的生命周期：程序的生命周期。

1.3.4 常量的分类

首先，字面常量，各种类型的已知的量，如1、’a’等

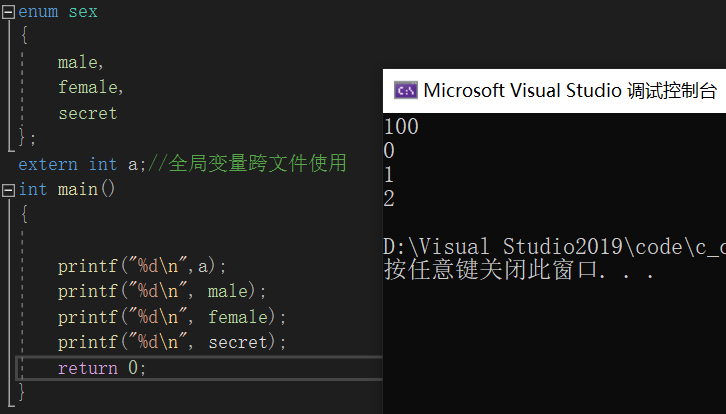
然后，const修饰的常变量，不能被修改的变量（本质是变量）。

const int a=0;a=20;//错误，不能被修改

const int n=10;int a[n]={0};//错误，本质是变量，不能用来定义数组。

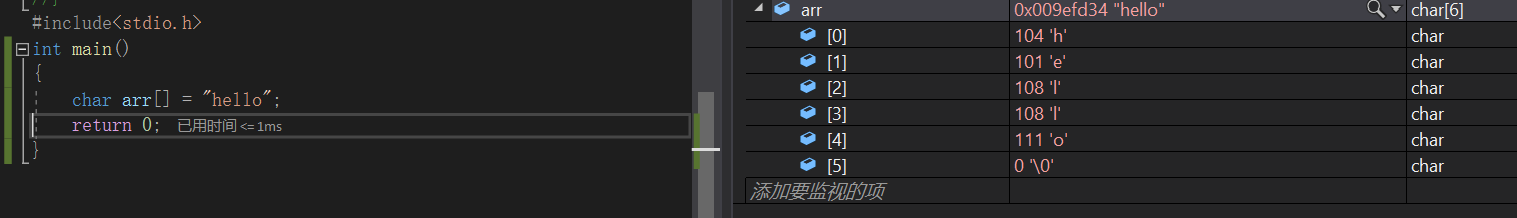
其次，#define定义的标识符常量，#define max 1000

最后，枚举常量enum，结构类似结构体，每个元素对应的值默认从0开始。用于可以一一列举的常量，比如星期、性别。如：

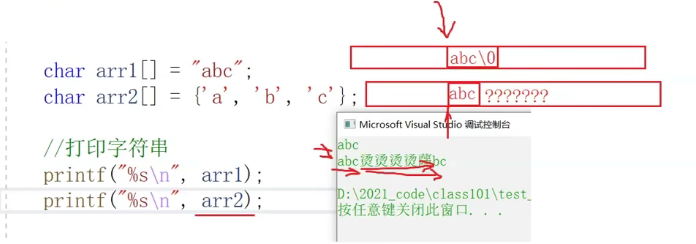


1.3.5 字符串、转义字符、注释

字符串就是一串字符——用双引号引起的一串字符，字符串的结束标志是\0的字符，不计入字符串长度strlen(arr)，遇见结束标志计算完毕，可运用字符数组存储。由调试可以看到字符串具有结束标志字符：



查看结束标志的作用：



#include<stdio.h>

#include<string.h>

int main()

{

char arr1[] = "abc";

char arr2[] = { 'a','b','c' };

printf("%d\n", strlen(arr1));

printf("%d\n", strlen(arr2));

return 0;

}

转义字符\，转变了它原来的意思。\n换行，\t制位



键盘上敲出的字符，如@#等，每一个字符都有一个值，ASCII值，ASCII表：



printf("%c\n", 65);//ASCII码转换

printf("%d\n",'a');

printf("%d\n",strlen("c:\test\328\test.c"));//14 \32为八进制转义，所以后面的8不是转义的整体

注释是用来解释代码的，注释方法：1、//C++注释风格2、/\* \*/C注释风格，不支持嵌套。

1.4 语言结构

1.4.1 选择结构

#define \_CRT\_SECURE\_NO\_WARNINGS 1

#include<stdio.h>

#include<string.h>

int main()

{

int a = 0;

printf("你要好好学习吗？0/1\n");

printf("你的选择是：");

scanf("%d",&a);

if (a == 0)

printf("回家喂猪\n");

else

printf("祝你成功！！\n");

return 0;

}

1.4.2 循环结构

例：抽奖和好好学习

#define \_CRT\_SECURE\_NO\_WARNINGS 1

#include<stdio.h>

int main()

{

int i = 0;

int c = 0;

srand((int)time(0));//避免代码两次运行产生相同的随机数

for (i = 0; i <= 1000; i++)

{

c = rand()%100;

printf("c=%d\n",c);

if (c == 1)

{

printf("恭喜你中奖了\n");

break;

}

if(i<1000)

{

printf("i=%d\n",i);

printf("人丑多读书\n");

}

else if (i == 1000)

{

printf("i=%d\n",i);

printf("恭喜你成为大神\n");

}

}

return 0;

}

1.5 函数

数学函数f(x,y)=2\*x+3\*y，参数及返回值。

#include<stdio.h>

int main()

{

int x = 0;

int y = 0;

int s = 0;

printf("请输入两个数\n");

scanf("%d%d",&x,&y);

s = sum(x, y);

printf("两个数的和为sum=%d\n",s);

return 0;

}

int sum(int x, int y)

{

int sum = 2\*x + 3\*y;

return sum;

}

1.6 数组

数组是一组相同类型元素的集合，下标。定义变量开辟空间。

1.7 操作符

第一，算术操作符，+ - \* / %

第二，移位操作符，<<左移，>>右移，移位顾名思义为移动二进制位，左移低位补零，右移高位补零 注意数据类型所占空间大小。

第三，位操作符，&按位与，|按位或，^按位异或

第四，赋值操作符，复合赋值操作符。



如b+=2;等价于b=b+2;同理，b>>=2;等价于b=b>>2;b&=2;等价于b=b&2;

#include<stdio.h>

int main()

{

int b = 2;

b += 4;//6

printf("%d\n", b);

b &= 2;//0110&0010=0010=2

printf("%d\n", b);

b <<= 2;//8

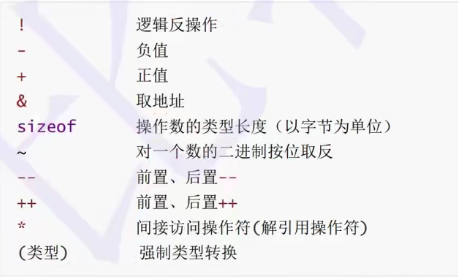
printf("%d\n", b);

return 0;

}

第五，单目操作符，即只需要一个操作数即可进行操作的操作符。

1.！a逻辑反（非零或零或一，真假逻辑切换，条件语句）。



2.Sizeof是一个操作符，括号在计算变量空间大小时可以省略。也可以计算数组内存大小以及数组所含元素个数。

#include<stdio.h>

int main()

{

int b = 0;

int arr[10] = { 0 };

int sz = 0;

printf("%d\n", sizeof b);

printf("%d\n", sizeof(int));

printf("%d\n", sizeof(arr));//计算数组所占内存大小，每个元素为整型

printf("%d\n", sizeof(arr[1]));//计算数组中某个元素所占大小，整型

sz = sizeof(arr) / sizeof(arr[1]);

printf("%d\n", sz);

return 0;

}

3.按位取反操作符~，注意负数的二进制写法。整数在内存中是按补码存放的，正数的原码、反码、补码与原码相同；负数的原码为最高位为1（符号位）其余位为绝对值的二进制（由此负数能表示的范围低于正数）。负数的反码是最高位为1，其余位取反。负数的补码是反码再加一个最低位的1.在计算机中，负数是以原码的补码形式表达的。所以要确定一个负数的值，必须要把它转换成原码。如负数11111111 11111111 11111111 11111011在计算机中存储，其真实值为补码减最低位1，再取反，得10000000 00000000 00000000 00000101=-5.

int b = 2;

printf("%d\n", ~b);//取反得到计算机中得存储形式，再转换为-3

4.--，++，前置与后置，前置先运算在使用，后置是先使用后运算。

5.( )强制类型转换int a = (int)3.14;

6.\* 、&

第六，关系操作符，>,<,!=,==;

第七，逻辑操作符，&&逻辑与，||逻辑或，if(a&&b)

第八，条件操作符，三目操作符。exp1?exp2:exp3,exp1成立，exp2计算并且为结果；exp1不成立，则exp3计算并且为结果（表达式）。

int max = 0;

max = a > b ? a : b;//找出a,b中的最大值

第九，逗号表达式，逗号隔开得一段表达式，从左向右以次计算，整个表达式得结果为最后一个表达式的结果。

int a = 5;

int b = 2;

int c = 1;

int d = 0;

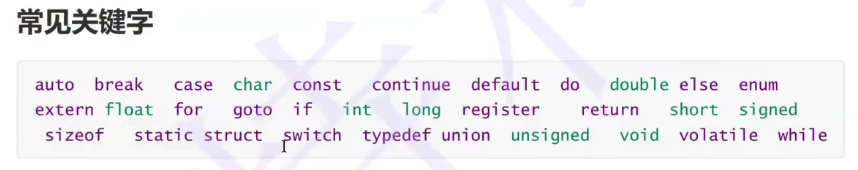
d = (a = b + c, b = a - 2, c = b + 3);//逗号表达式3 1 4 4

printf("%d\t%d\t%d\t%d\n", a, b, c, d);

第十，下标引用[]、函数调用( )、结构体访问 . 点操作符或->

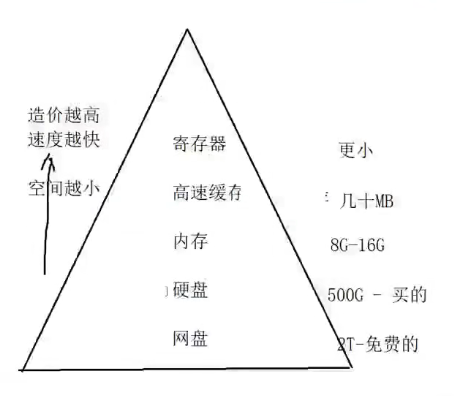
1.8 常见关键字

特点1.C语言提供的，不能自己创建关键字。2.变量名不能是关键字



1.auto，局部变量都是自动创建自动销毁的，自动变量，在定义时省略了。

2.register，寄存器关键字，建议把定义的变量存放到寄存器中register int num = 0;计算机中数据可以存到寄存器（很小内存）、高速缓存（几百M）、内存（8，16g）及硬盘（500g）。造价越高速度越快，但空间越小。为了使CPU和内存的速度尽量同步。



3.signed 有符号的，unsigned 无符号的

4.static 静态的

5.union 联合体

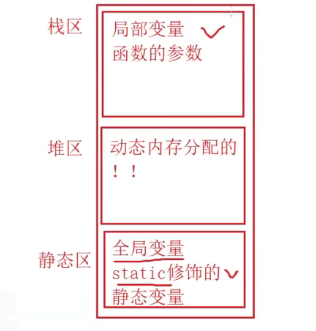
6.typedef数据类型定义

define 、include 不是关键字，预处理指令

typedef 类型重定义关键字 typedef unsiged int u\_int//将unsiged int数据类型重新定义为u\_int，复杂类型名简化。

Static静态的：

（1） Static修饰局部变量：使得局部变量在出局部范围之后变量不销毁，改变了变量的生命周期（本质上改变了变量的存储类型）



（2） Static修饰全局变量：使得这个全局变量只能在自己所在的源文件使用，不能在其他源文件中使用。全局变量之所以能被其他源文件使用，是因为全局变量具有外部属性，但是在被Static修饰之后，就变成内部链接属性了，这样其他源文件就不能链接到这个静态的全局变量。

（3）修饰函数：声明函数extern int sum(int x, int y);外部源文件的函数应用。Static修饰函数使得函数只能在自己的源文件内部使用，不能在其他源文件使用，依然改变了函数的外部链接属性，这与Static修饰全局变量一致。

#define定义常量和宏：#define是一个预处理指令，主要两个用途：第一，定义符号常量；第二，#define定义宏，宏具有参数，#define定义的宏主要是实现替换效果，如：

#include<stdio.h>

#define add(x,y) x+y

int main()

{

int x = 2, y = 3;

printf("%d\n", 4\*add(x, y));//add(x,y)替换成x+y，即这里为4\*x+y=11

return 0;

}

要正确使用宏定义，则加上括号#define add(x,y) ((x)+(y))才正确

1.9 指针（内存）

1.9.1 前言

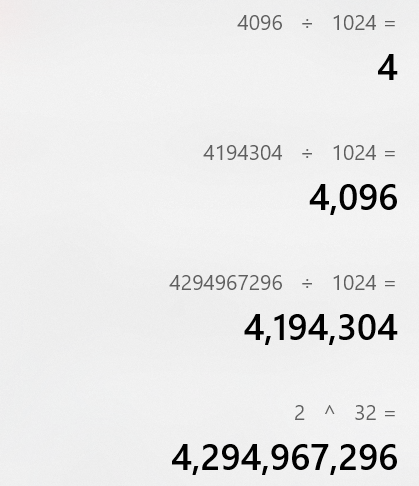
内存即空间，空间都是被划分为一个一个的小空间，如何访问这些空间呢，就是按照地址访问的，如我们每个人的寻找是按地址来访问的。这就引出了两个问题：

（1）内存是如何编号的呢？

64位、32位：为32跟地址线-物理线-通电-电信号转换为数字信号，0和1的二进制序列，32位二进制地址线来对地址进行编号，产生2^32个地址编号，这32位编号就为内存的地址。

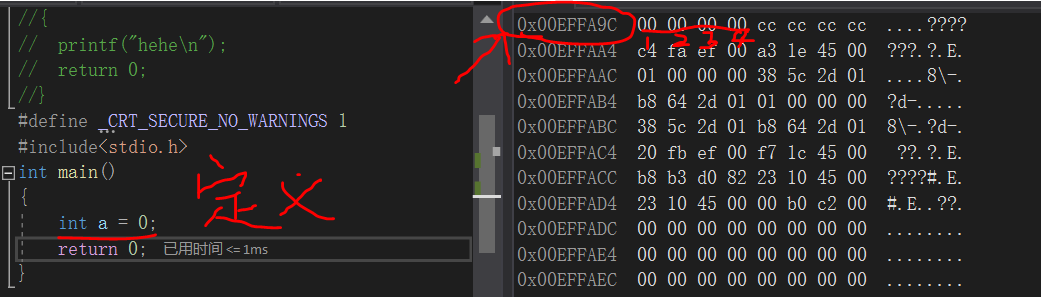
（2）一个这样的内存单元是多大空间呢？

按照权衡，定义一个内存单元的大小为一个字节，那么32位地址线就可以编号2^32个字节的空间，即空间的大小为2^32byte，最终化为空间为4G内存，那么64位则为8G空间。



1.9.2 指针

当我们创建一个变量之后，计算机就相应的安排一个指定地址的内存空间，如int型数据占四个字节，那么它的地址是指其首地址（&），空间内部放的是变量的值。



printf("%p\n",&a);//%p专门用来取地址，程序每次运行重新分配地址。

int \*pa = &a;//pa是用来存放地址的变量，在C语言中称为指针变量

//\*代表pa是指针变量，int表明指针所指向变量为int类型。

char b = 'w';

char \*p = &b;

指针变量：用来存放内存地址的变量。

存放地址的目的是为了找到该地址的内容，\*p表示指针p所指向的内容，可以间接改变内存中的内容，\*解引用操作符。指针变量存的是地址，地址指向变量。

1.9.3 指针变量的大小

sizeof(char \*);指针变量存的是地址，所以指针变量的大小由地址的大小决定，如32位地址线对应指针的大小为4个字节（X86），64位地址线对应的是8个字节（X64）这是固定的，指针的大小与指针的数据类型无关。

1.10 结构体（使得C语言具有描述复杂类型的能力）

主要三个步骤：

（1）定义结构体类型（2）结构体类型定义变量（3）变量引用‘.’操作符

//定义结构体类型

struct stu

{

char name[20];

int age;

double score;

};

//用类型定义变量变量初始化

struct stu s= { "张三",20,85.5 };

//数据引用

printf("%s %d %lf", s.name, s.age, s.score);

‘->’操作符，结构体指针指向结构体成员

struct stu\* pa = &s;

printf("%s %d %lf", pa->name, pa->age, pa->score);

初始C语言结束，在线OJ刷题，必须要按照题目要求，不能多加效果等。

#include<stdio.h>

#include<string.h>

int main()

{

struct student

{

char name[20];

float ch;

float en;

float ma;

float avg;

};

struct student s1 = { "张三儿",90.5, 88.0, 66.5 };

s1.avg = (s1.ch + s1.en + s1.ma) / 3;

printf("%s的平均分为%f\n", s1.name, s1.avg);

return 0;

}

#include<stdio.h>

#include<string.h>

int main()

{

union student

{

float ch;

float en;

float ma;

float avg;

};

union student s1;//联合体每次只能访问一个

struct stu

{

float ch;

float en;

float ma;

float avg;

};

struct stu s2;

printf("%d\n%d\n", sizeof(s1), sizeof(s2));

return 0;

}