* **e或E之前必须有数字，且e或E后面必须为整数，如12.34e-3。**
* **C语言本身不提供输入输出语句。**
* **char a[] = "\\TOP\t\65\""; // 结果：\TOP 5"**

**【注】长度为7,大小为8.**

* **逗号运算符的运算顺序是自左向右的, 如y=((x = 4 \* 5, x \* 5),x + 25)结果是45.**

**a = 2 \* 6, a \* 3, a + 5; // 17**

**【注意括号作用范围,不在括号内赋值运算优先计算，结果就是第一个的值】**

* **运算符和结合性：**

**-i++ 等价于 -(i++) 结合顺序自右向左,++优先级高,故输出结果就等于-i**

**i---i等价于i-- -i 结果为0**

**int (\*p)[4]表示指向一维数组的指针,而int \*p[4]表示指针数组**

**\*p++和(\*p)++ 意义不同,前者表示下标加一,后者表示p指向的数值加一**

**b -=b += b\*b 等价于b=b-(b=b+(b\*b)) 结果等于0**

* **(int)x强制转换只保留整数部分,如8.2,8.9结果都是8**
* **float x = 213.82631; printf(“%3d”, (int)x);其中的3d表示输出占三位(213),若为4d表示占四位( 213),若所占位数不足时,依然按照原数的长度输出,如2d结果还是213.**
* **字母大小写转换：’a’-32 == ‘A’ ‘A’+32==‘a’**
* **判断文件打开是否成功：if((fp=fopen(filename,’r’)==NULL) printf(“文件打开失败”);**
* **程序文件：源文件（.c）、目标文件（.obj）、可执行文件（.exe）**
* **数据文件：ASCII文件（文本文件）和二进制文件（映像文件）**
* **C语言程序结构有以下特点：**

1. **一个程序由一个或多个源程序文件组成。**
2. **函数是C语言的主要组成部分**
3. **一个函数包括两个部分**
4. **程序总是从main函数开始执行的**
5. **程序中对计算机的操作是由函数中的C语句完成的**
6. **在每个数据声明和语句的最后必须有一个分号**
7. **C语言本身不提供输入输出语句**
8. **程序中应当包含注释**

* **运行C语言步骤和方法：**

**（1）输入和编辑源程序**

**（2）对源程序进行编译**

**（3）进行连接处理**

**（4）运行可执行程序**

* **程序=数据结构+算法**
* **算法分为：数值运算算法和非数值运算算法。**
* **算法的主要特征：有穷性、确定性、有零个或多个输入、有一个或多个输出、有效性。**
* **算法常用表示方法：自然语言、传统流程图、结构化的流程图、伪代码。**
* **圆角矩形表示开始、结束；平行四边形表示输入输出框；菱形框表示判断；普通矩形表示处理；箭头表示流程线；圆表示连接点。**
* **三大基本结构：顺序、选择、循环结构。**
* **判断素数：**

**int prime(int n)**

**{**

**int i, k= sqrt(n);**

**for(i=2; i<=k; i++)**

**{**

**if(n%i==0)**

**return 0;**

**}**

**if(i>k)**

**return 1;**

**else**

**return 0;**

**}**

* **求最大公约数最小公倍数：**

**int gcd(int a, int b)**

**{**

**if (b == 0)**

**return a;**

**else**

**return gcd(b, a%b);**

**}**

**int lim(int a, int b)**

**{**

**return a \* b / gcd(a, b);**

**}**

* **采取以下方法得到结构化的程序：**

**（1）自顶向下**

1. **逐步细化**
2. **模块化设计**
3. **结构化编码**

* **符号常量不占内存，只是一个临时的符号。**
* **数据类型：**

1. **基本类型：**
   * + - **整形类型：int,short int,long int,long long int,char,bool.**
       - **浮点型：float,double,float\_complex,double\_comple,long long\_comple.**
2. **枚举类型：enum**
3. **空类型：void**
4. **派生类型：指针类型、数组类型、结构体类型、共用体类型、函数体类型。**

* **无符号整形数据用”%u”格式输出。**
* **自增，自减只能用于变量，不能用于常量或者表达式。如5++,（a+b）++均不合法。**
* **puts（输出字符串），gets（输入字符串），getchar（输入字符），putchar（输出字符）.**
* **“%5d” 表示数据占五列**
* **“%o”输出八进制**
* **“%x”输出十六进制**
* **printf("%13.2e",123.456); // 1.23e+002前面四个空格**
* **printf(“%5.2f”,a);//表示数据宽度为5,小数位数为2**
* **putchar(‘\101’); // A**
* **putchar(‘\’’); // 单撇**
* **putchar(‘\015’); // 回车**
* **printf(“%d”,’a’);**
* **运算符优先顺序：(低→高)**

**逗号运算符🡺赋值运算符🡺条件运算符🡺||🡺&&🡺|🡺^🡺&🡺!=、==🡺关系运算符🡺位移运算🡺算术运算符🡺sizeof、取地址、指针、负号、自减、自加、按位取反、逻辑非🡺结构体成员运算符、指向结构体成员的运算符、下标运算符、括号**

**如：a>b==c ⬄(a>b)==c**

**!a+b || a=b>=1 && b+=c%d ⬄ ((!a) + b) || ((a=(b>=1)) && (b+=(c%d)))**

**相同优先级中，按结合顺序计算。大多数运算是从左至右计算，只有三个优先级是从右至左结合的，它们是单目运算符、条件运算符、赋值运算符。**

* **基本的优先级需要记住：**
* **指针最优，单目运算优于双目运算。如正负号。**
* **先乘除（模），后加减。**
* **先算术运算，后移位运算，最后位运算。请特别注意：1 << 3 + 2 & 7等价于**

**(1 << (3 + 2))&&7.**

* **逻辑运算最后计算。**
* **getchar输入两个字符时不能有空格。**
* **输入双精度浮点数用“%lf”。**
* **条件表达式：a > b ? a : b //当a>b时，输出a；否则输出b。**
* **判断闰年：(year % 4 == 0 && year % 100 != 0) || (year % 400 == 0)**
* **判断浮点数等于0用fabs(a) <= 1e-6**
* **for（表达式1，表达式2，表达式3）{ 语句 } 循环执行过程：先执行表达式1，再求解表达式2，若真则继续执行循环体语句，否则结束循环，接下来执行表达式3.**
* **数组中的元素都属于同一种数据类型。**
* **在main函数中不允许定义（动态数组）int a[n],但是在调用函数中可以,如下所示：**

**int fun(int n)**

**{**

**int a[n]; //若改为static int a[n]，则此处是错误的。**

**...**

**}**

**【注】因为调用函数是临时分配内存的，故每次调用时数组大小可以改变。**

* **char a[ ] = "china"; ⬄ char a[ ]={‘c’,’h’,’i’,’n’,’a’,’\0’};//也可以char a[]= {"china"}; 系统处理字符串常量时会在末尾自动添加’\0’，此时数组长度为5，大小为6。**
* **用新字符串“hello”赋值原字符串“C program”。结果为hellogram，原因是没有加’\0’结束符。**
* **如果数组长度大于字符串实际长度，也只输出到遇到’\0’结束。**
* **puts(str); //会将结束标志’\0’转换成’\n’。因此自动换行。**
* **puts(str)，gets(str)只能输入输出一个字符串。//不允许gets(str1,str2)。**
* **strcpy(str1,”china”)；第二个参数可以为字符串或数组名，第一个必须为数组名。**
* **str1=”cabc”; str1=str2; //不能直接赋值，都是非法的。**
* **strncpy(str1,str2,2);将str2中的前两个字符复制到str1。**
* **strcmp(str1,str2);比较时按ASCII码大小比较。如“DOG”<“cat”。**

**结果相等时为0，大于为正数，小于为负数。**

* **大写字母转小写：ch+32; 小写转大写：ch-32。**
* **strlwr(str)； //将字符串转小写**
* **strupr(str)； //将字符串转大写**
* **一个c程序由一个或多个程序模块组成，每个程序模块作为一个源程序文件。**
* **一个源程序文件由一个或多个函数以及其他有关的内容组成。**
* **C语言执行程序从main函数开始，也是从main函数结束。**
* **函数不能嵌套定义。**
* **函数体包括声明和语句部分。**
* **形参和实参类型应该相同或赋值兼容。**
* **实参向形参的数据传递是“值传递”，单向传递，只能由实参传给形参，而不能形参传给实参。它们在内存中存放的位置不同。**
* **函数类型决定返回值类型。**
* **float add(int a,int b); //函数声明。**
* **int fun(int, float, double);⬄** **int fun(int a, float b, double c);**
* **一维形参数组可以指定大小或不指定大小，两者本质上一样（编译系统不检查其大小，只传递数组首地址），但是二维数组列数必须一致，行数可不一致。**
* **变量定义的三种情况:**
* **在函数的开头定义**
* **在函数的复合语句内定义。**
* **在函数的外部定义。**
* **汉诺塔三步骤：**
* **将A上的n-1个盘借助C座先移到B座上。// hanoi(n-1,one,three,two);**
* **将A座上剩下的一个盘移到C座上。// move(one,three);**
* **将n-1个盘从B座借助A座移动到C座上。// hanoi(n-1,two,one,three);**
* **全局变量的缺点：**
* **全局变量在整个程序执行过程中都要占用内存。**
* **全局变量降低函数的通用性。**
* **全局变量降低程序的可读性。**
* **全局变量和局部变量同名时：在局部变量的范围内，全局变量被屏蔽。**
* **变量的存储有两种：静态存储方式和动态存储方式。**
* **存储空间：**
* **程序区**
* **静态存储区（存放全局变量）**
* **动态存储区（函数形式参数、函数定义声明的自动变量、函数调用时的保护现场和返回地址）**
* **C语言的存储类别：自动的（auto）、静态的（static）、寄存器的（register）、外部的（extern）。**
* **每个变量和函数都有两个属性：数据类型和存储类别。**
* **静态局部变量在编译时赋初值，且只赋初值一次，在程序运行时已有初值。其在整个运行过程中不释放。**
* **用static将外部变量的作用域限制在本文件中。**
* **正确而灵活地使用指针，可以使程序简洁、紧凑、高效。**
* **指针变量只能存放地址，不能将一个整数赋值给它。**
* **p[2]并不一定等于a[2],只有当p指向a[0]时才会等。**
* **字符指针变量指向的字符串常量中的内容是不可以改变的。如：char \*p=”Helolo”; p[2]=’r’; //非法**
* **引用数组各元素的值三种方法：下标法、数组名计算数组元素地址、指针变量指向数组。**
* **数组名a是指针常量，无法实现a++。**
* **p[3]=(void ) &a; //p3只是得到了a的纯地址，而没有指向a，故不能输出\*p3。**
* **\*（p++） ⬄ \* p++ //先取值，后++。**
* **\*（++p）//先++，再取值。**
* **++（\*p）//p指向元素加一**
* **\*（a[i]+j）⬄ \*（\*（a+i）+ j）//表示a[i][j]**
* **&a[i]或a+i指向行，而a[i]或\*（a+i）指向列。**
* **int (\*p)[4] //表示定义一个指针变量，它指向包含四个整型元素的一维数组。**
* **int \*p[4] //表示指针数组**
* **若int a[4]，int (\*p)[4]，则p = &a; //表示指向一维数组（行）。写成p=a是错误的，这样写表示p的值是&a[0],指向a[0]。**

**此时的（\*p）[3] ⬄ a[3]。**

* **char \*str=”adnd” 的大小为5，最后一个字符存放’\0’,str指向字符串的第一个字符。**
* **char str[]={‘a’,’d’}; //长度为8(此处长度有时不定)，大小为2.**
* **char \*p; \*p=”dumin”; //这是错误的,应为p=”dumin”;**
* **特别提示：字符串数组复制时最后要记得加结束标志’\0’。**
* **字符串题目易错点：**

**char p[]="chinese",q[]="dumin";**

**char \*a = p, \*b = q;**

**cout << a << b<<endl;**

**while ((\*(a++) = \*(b++)));**

**cout << a << endl; //结果：e,因为a已经指向了第七个字符的位置了。**

**cout << p << endl; //结果：dumin**

* **char str[14]; str=”dhhd”;//错误，数组名为常量。**
* **指向函数的指针：int (\*p)(int,int); p = max; c = (\*p)(a,b);**

**【注】：函数的指针类型是int（\*）（int，int）**

* **void \*malloc(unsigned int size) //动态分配一个长度为size的连续空间**
* **void \*callmoc(unsigned n, unsigned size)//分配n个长度为size的连续空间**
* **void \*free(void \*p)//释放空间**
* **void \*realloc(void \*p,unsigned int size)//重新分配大小**
* **int a=3; int \*p1=&a; void \*p; p=(void \*)p1; cout<<\*p<<endl; //错误，p为纯地址，不能指向a。**
* **struct Student{**

**int num;**

**char name[10];**

**}s[3] = { { 1, " " }, { 1, " " }, { 1, " " } }; //也可s[3] = {1," ",1," ",1," "};**

**Student \*p;**

**p = s; //等价于p = &s[0];**

**cout << p->num << (\*p).num << endl; //此处不可以写成 \* p.num**

* **共用体：几个变量共享一个内存段。**

**union Student{**

**int i;**

**double d;**

**float f;**

**char c;**

**};**

**Student p = {90};**

**cout << p.i <<” ”<< p.c << endl; //结果：90 Z**

**共用体类型特点如下：**

* **同一个内存段可以用来存放几种不同类型的成员，但同一时刻只能存放一个。**
* **可对共用体变量初始化，但只能初始化一个。**
* **共用体变量中起作用的是最后一次被赋值的成员。**
* **共用体变量的地址和他的各成员的地址都是同一个地址。**
* **不能对共用体变量名赋值，也不能企图引用变量名得到一个值。**
* **以前的c规定不能用共用体作函数的参数，但是c99允许。**
* **共用体类型可以出现在结构体类型的定义中，也可以定义共用体数组。反之也可。**
* **枚举类型：**

**enum { sum, monday, tue, wed, thu }week; //默认值为从0开始，依次加1。**

**【注】若初始时monday=2，则sum=0, tue=3, wed=4, thu=5。**

**若初始时sum=1, monday=3，则tue=4, wed=5, thu=6。**

**week = monday; //变量赋值**

**monday = 1; //错误，枚举元素是常量不能被赋值。**

* **typedef类型使用：（相当于别名）**
* **typedef int Integer;//此时Integer 等同于int**
* **typedef struct {**

**int num;**

**int name;**

**}Data;**

**Data p;**

* **typedef int Num[100];**

**Num a;**

* **typedef char \* String;**

**String p, s[10];**

* **typedef int (\*Pointer)();**

**Pointer p1,p2;**

* **程序文件：源文件（.c）、目标文件（.obj）、可执行文件（.exe）**
* **数据文件：ASCII文件（文本文件）和二进制文件（映像文件）**
* **fopen(“shiyan”,”r”); //文件名,打开方式**
* **常见文件打开方式：**

**“r”只读，若文件不存在，则出错。**

**“w”只写，若文件不存在，则建立新文件。**

**“a”追加，若文件不存在，则出错。**

**“rb”只读，若文件不存在，则出错。**

**“wb”只写，若文件不存在，则建立新文件。**

**“ab”追加，若文件不存在，则出错。**

**“r+”读写，若文件不存在，则出错。**

**“w+”读写，若文件不存在，则建立新文件。**

**“a+”读写，若文件不存在，则出错。**

**“rb+” 读写，若文件不存在，则出错。**

**“wb+” 读写，若文件不存在，则建立新文件。**

**“ab+” 读写，若文件不存在，则出错。**

**【注】记忆方法：带“w”的才会创建新文件，带“b”的是表示二进制文件，**

**带“+”的表示读写均可。**

* **文件读写：**

**FILE \*in,\*out;**

**if ((in = fopen("a.txt", "r")) == NULL)**

**{**

**printf("无法打开文件！");**

**exit(0);**

**}**

**if ((out = fopen("b.txt", "w")) == NULL)**

**{**

**printf("无法打开文件！");**

**exit(0);**

**}**

**char ch;// //读取单个字符**

**while (!feof(in)) //如果没有遇到文件结束符**

**{**

**ch = fgetc(in); //等同于getc(in)**

**fputc(ch, out); //等同于putc(in)**

**}**

**char str[5]; //读取字符串**

**while (fgets(str, 5, in)!=NULL)**

**{**

**fputs(str,out);**

**}**

**fclose(in);**

**fclose(out);**

* **fread(buffer,size,count,fp)；**
* **fwrite(buffer,size,count,fp)；**
* **fputs(char \*s,fp);**
* **fputc(char s,fp);**
* **fgets(char \*s,int n,fp);**
* **fgetc(char s,fp);**
* **fprintf(fp,”%d”,a); //文件指针、格式化、数据**
* **fscanf(fp,”%d”,a); //文件指针、格式化、数据**
* **用二进制的形式向文件读写一组数据的范例代码：**

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

**{**

**scanf("%d%lf",s[i].num,s[i].score);**

**fwrite(&s[i],sizeof(Student),1,fp); //表示向文件读入1个Student的数据。**

**fread(&s[i],sizeof(Student),1,fp); //表示从文件读取1个Student的数据。**

**}**

**fclose(fp);**

* **rewind（fp）; 使文件指针重新指向文件头。**
* **fseek(文件类型指针，位移量，起始点) //起始点0表示文件开始位置，1表示文件当前位置，2表示文件的末尾位置。**

**示例如下：**

**fseek(fp,100L,0); //表示移到离文件文件开头（之后）100个字节的位置。**

**fseek(fp,100L,1); //表示移到离文件文件当前位置100个字节的位置。**

**fseek(fp,-100L,2); //表示移到离文件文件结尾（之前）向后100个字节的位置。**

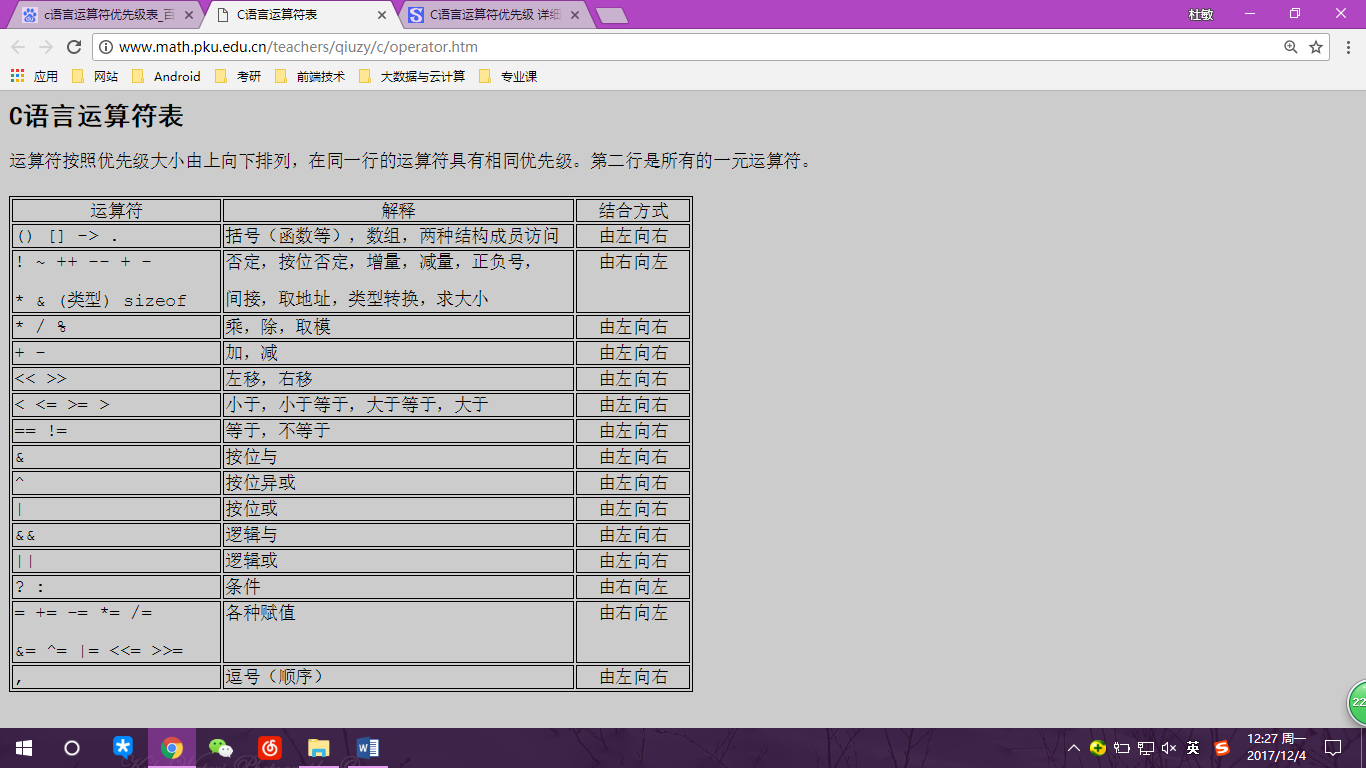
**【注】L表示long 型数据。第二个参数为正数表示起始点向后的偏移量，负数表示偏移量向前的偏移量。**

* **ftell(fp); //测定文件位置标记的当前位置。**
* **ferror(fp); //返回为0，表示为出错；否则出错。**
* **int a(10); //表示对a赋值**
* **文件指针指向指针，而文件内部指针则是用于标记当前读写位置。两者意义不同。**
* **scanf("a=%db=%dc=%d", &a, &b, &c); //此时输入为：a=1b=2c=3 【不能有空格】**
* **结构体中：如struct Student s1,\*s2; //则只能写成s1.a或s2→a或 (\*s2).a**
* **文件打开(关闭)含义：建立（撤销）相应的信息区和文件缓冲区。**
* **指向文件的指针变量并不是指向外部介质上的数据文件的开头，而是指向内存中的文件信息区的开头。**
* **宏定义后面不加分号。#define PI 3.14**
* **宏名与带参数的括号之间不能有空格。#define S(r) PI\*r\*r a=S(3);**
* **int a[][4]; //单纯的这样申明是不对的。**

**【必须有数据初始化，如int a[][4]={1,2,3,4,5,6};】**

* **逻辑运算：**
* **&（与）：1&1=1,其他情况均为0.**
* **|（或）：有1则结果为1，否则为0.**
* **^（异或）：同号为0，异号为1.**
* **~（取反）：0变1, 1变0.**
* **≫（左移）：乘以2.**
* **≪（右移）：除以2.**

C语言运算符附表

**参考网址：**[**http://blog.csdn.net/u013630349/article/details/47444939**](http://blog.csdn.net/u013630349/article/details/47444939)