考试大纲2010-2019





试题难度：1:3:6

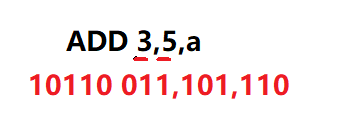
第一章 程序设计和C语言

1.程序：就是一组**计算机**能识别和执行的**指令**。

2.计算机语言

**机器语言：**就是由0和1组成的计算机可以识别的语言。

**符号语言（汇编语言）：通过符号表示二进制数。**



**缺点：不通用**

高级语言：使用很接近于人类的语言完成计算机程序的设计。

特点：接近人类，不依赖机器。

高级语言-🡪编译程序🡪机器语言

第一高级语言：FORTRAN

高级语言的发展：非结构化(BASIC)->结构化-(C语言)->面向对象(Java,C#,C++,Python,PHP)

C语言的标准：C89和C99

C语言的注释：//行注释 /\* \*/段注释（C99）C89---》/\* \*/段注释

C语言特点：面向过程，可以应用系统程序和应用程序的开发。

|  |
| --- |
| /\*  这是第一个C语言程序  \*/  #include<stdio.h>//头文件 std标准 io输入输出  int main(){//主函数--程序的入口main  //int--->返回值  printf("Hello!");//输出语句，表示打印Hello  return 0;//返回一个0,表示正常结束；异常结束返回一个  //非0值  } |

注意：每一个C语言程序**必须**有一个main()函数。

|  |
| --- |
| #include<stdio.h>  int main(){  //程序-->内存 --->执行  int a,b,sum;//定义变量,在内存中分配空间  a = 100;  b = 30;//给变量赋值  sum = a+b;  printf("你好sum=%d\n",sum);  //"sum=%d\n"输出格式  //sum输出的变量 %d表示输出一个整数  //\n回车换行  printf("hello");  return 0;  } |

C语言的程序结构

1.一个C语言程序是由一个或者多个**源程序**组成的。

2.一个C语言程序是由一个或者多个函数构成的，但必须包含main()函数，并且只能有一个main()。

3.C语言程序总是从main()函数开始执行的。

4.每个**C语言语句**后必须有一个分号。

a = 100;---🡪c语言语句

a = 100--🡪c语言表达式

5.C语言本身不包含输入(scanf)和输出(printf)语句,是通过stdio.h头文件引入的。

写C语言程序的步骤和方法

**源程序---🡪 .C文件 .CPP文件**

（1）编写c语言的源程序（.c或.cpp），数据文件。

（2）对源程序进行编译（目的：检查程序是否有错误，将源程序转化为二进制文件）-🡪目标程序。

（3）对编译后的文件进行连接处理。（将多个编译后的C语言文件进行组合）

（4）可以得到可执行文件（.exe）

第二章 算法

1.程序=算法+数据结构

（1）对数据的描述为数据类型和数据的组成形式，称为数据结构。int a;自定义的数据类型 class Student{。。。。}

（2）算法：解决问题的步骤和方法。

做菜（1）准备材料（数据）（2）制作步骤（算法）

计算机的算法：数值运算算法（10%）和非数值运算算法（90%）。

算法的特性：

（1）有穷性：算法的步骤是有穷的。

（2）确定性：算法的步骤是确定的。

（3）有**零个**或者多个输入：输入必要的数据。

（4）有一个或者多个输出：算法的“解”。

（5）有效性：算法的每一个步骤都是可以有效执行的。

如何描述一个算法？

（1）用自然语言表示算法

（2）用程序流程图

（3）N-S流程图表示算法

（4）用伪代码表示算法

（5）用计算机语言表示算法

C语言的三种基本结构？

（1）顺序结构

（2）选择结构（if和switch）

（3）循环结构（while和for）

第三章 顺序程序设计

1.数据的两个形式：常量和变量。

（1）常量：在程序运行时，值不能变化的量。

整型常量：100，-2,200；

实型常量：（1）十进制表示：1.2 23.5

（2）指数表示：12.2e-2

字符常量：（1）普通字符：’a’ ‘?’

（2）转义字符：\n 表示回车换行

\t 表示光标移动到下一个tab的位置。

[\\表示字符\](file:///\\\\表示字符\\)

‘\101’ ‘\x41’

字符串常量：”avc” “a”

注意：”a”和’a’的区别？

符号常量

|  |
| --- |
| #include<stdio.h>  #define PI 3.1415 //符号常量  int main(){  int r=3;  r = 10;  printf("圆的面积：%f\n",PI\*r\*r);  printf("PI=%f\n",PI);    const int a = 10; //常变量  printf("a=%d\n",a);  return 0;  } |

**常变量**和**符号常量**有什么区别？

常变量在程序编译时，需要占用内存，符号常量在编译时，不需要占用内存。

变量：使用方法：先定义，后使用。

int:整型 int a;在内存中开辟空间。

**Dev-c++：4Byte**；TurboC：2Byte；

64位机器：8Byte。

整型：

|  |
| --- |
| #include<stdio.h>  int main(){  int a;//定义变量，在内存中开辟4Byte大小  a = 10;//给变量赋值  //1Byte = 8bit  //4Byte = 32bit  //00000000 00000000 00000000 00000000  //11111111 11111111 11111111 11111111  //01111111 11111111 11111111 11111111  //1111=+7  //0111=-7  printf("a=%d\n",a);  short b = 12;//短整型，在内存中开辟2Byte  printf("b=%d\n",b);  long c = 13;//长整型，内存中开辟4Byte  printf("c=%d\n",c);  long long d = 13;//双长型， 内存中开辟8Byte  printf("d=%d\n",d);  return 0;  } |

short<=int<=long<=long long

无符号数

|  |
| --- |
| #include<stdio.h>  int main(){  //补码  unsigned int a = 100;  //11111111 11111111 11111111 11111111  //00000000 00000000 00000000 00000000  //1111=15  //0000=0  printf("a = %d\n",a);  unsigned int b = -100;  printf("b = %d\n",b);  printf("b = %u\n",b);//%u无符号数的输出格式  //0110  return 0;  } |

字符型：

|  |
| --- |
| #include<stdio.h>  int main(){  char ch = 'a';//会在内存中开辟1Byte。  //字符在计算机存储的时候，存储的都是字符的ASCII码  printf("ch = %c\n",ch);  printf("ch ascii = %d\n",ch);  //计算'1'-1的结果？ 48  char ch2 = '1';  int a = 1;  printf("%d\n",(ch2-a));  return 0;  } |

程序：将小写字符a转化为大写。

|  |
| --- |
| #include<stdio.h>  int main(){  char ch = 'a';//97 A:65 =32  //小写字符的ascii码和大写字符的ascii码相差32  char ch2 = ch - 32;  printf("ch2 = %c",ch2);  return 0;  } |

浮点数

单精度浮点型：float，4Byte，小数有效位数是6位。

双精度浮点型：double，8Byte，小数有效位数是15位。

|  |
| --- |
| #include<stdio.h>  int main(){  float a = 12.34519872123F;  printf("a = %.8f\n",a);  double b = 12.34519872123;  printf("b = %.11f\n",b);  //12.4--double 12.4f--float  float c = 12.3456;  //将double的数赋值给float类型，有可能发生数据溢出  return 0;  } |

标识符(变量名称)：

**规则：只能由字母，数字，下划线**

**\_构成，第一个字符必须是字母或下划线。例如:int 2ac; int \_2abc;**

算术运算：+(加法,正号) -(减法,负号) \* / % 10%3=1 2%5=2

|  |
| --- |
| #include<stdio.h>  int main(){  int a = 17;  int b = 5;  int c;  c = a+b; //=赋值运算符 +算术运算符  //算术运算符的优先级高于赋值运算符  printf("c= %d\n",c);  c = a-b;  printf("c= %d\n",c);  c = a\*b;  printf("c= %d\n",c);  c = a/b;//17/5 = 3 int/int=int  printf("c= %d\n",c);  c = a%b;//2.3%2  printf("c= %d\n",c);  //%两边的值，必须是整型  //printf("c= %d\n",2.3%2.3);错误  return 0;  } |

++和--

a++：a = a +1;先使用a的值，然后a再加1.

++a：a = a + 1;先将a的值+1,然后再使用a的值.

|  |
| --- |
| #include<stdio.h>  int main(){  int a = 18 ;  int b;  //b = a++;  //1.先将a的值赋值给b=18  //2.a = 19  b = ++a;  //1.a=19  //2.b=19  printf("a = %d,b=%d",a,b);  return 0;  } |

算术运算符的优先级：先乘除，后加减，结合顺序：从左到右。

a=b=c=d=100;

不同类型数据间的混合运算。

1. 在+-\*/运算的两个数，其中有一个数为float或double类型时，运算结果为double类型，因为系统会将所有的float类型先转化为double类型，再进行运算。
2. 如果int类型与float或double类型进行运算，先把int类型和float类型转化为double类型，然后进行运算，结果为double类型。
3. 字符类型char与整型进行运算结果为整型。char与实数类型进行运算，结果为实数类型。

|  |
| --- |
| #include<stdio.h>  int main(){  int i = 3;  float f = 3.5f;  double d = 7.5;  double dd;//从左到右  //1.计算10+'a'=107;107+i\*f-d/3  //2. i\*f=10.5;107+10.5-d/3  //3.107+10.5=117.5;117.5-d/3  //4.d/3=7.5/3=2.5;117.5-2.5  //5.115.000000  printf("dd=%f",dd);  return 0;  } |

强制类型转化

缺点：有可能引起精度的丢失。

|  |
| --- |
| #include<stdio.h>  int main(){  float a = 12.8456f;  int b;  b = (int)a;  printf("b = %d",b);  return 0;  //(int)((double)(7/2)+3.5)  //1.(int)((double)3+3.5)  //2.(int)(3.0+3.5)  //3.(int)6.5  //4.6  //(int)((double)7/2+3.5)  } |

赋值语句=

结合顺序：从右向左。

|  |
| --- |
| #include<stdio.h>  int main(){  //int a,b,c=10;  //int a=b=c=10;错误  int a=10,b=10,c=10;  printf("a=%d,b=%d,c=%d",a,b,c);  return 0;  } |

复合赋值运算：+=,-=,\*=,/=,%=

a+=3;//a=a+3;

|  |
| --- |
| #include<stdio.h>  int main(){  int a = 6,b=3;  //a+=3; a = a+3;  a\*=b+3;//a = a \* (b +3)  printf("a=%d",a);  return 0;  //X-=X\*X  //X=X-(X\*X)=5-5\*5=-20  } |

赋值过程中的类型转化

1. 将浮点型数转化为整型，直接舍弃小数部分。
2. 将整型转化为浮点型，数值不变，以小数的形式保存。
3. 将一个double类型数据转化为float类型数据时，注意数值范围能溢出。
4. 字符型数据赋值给整型变量，将字符的ASCII码直接赋值。
5. 整型数据赋值给字符型数据，可以发生数据错误。

|  |
| --- |
| #include<stdio.h>  int main(){  //1.浮点型-->整型  float a = 12.5f;  int b;  b = a;//高精度--->低精度  printf("b=%d\n",b);  //2.整型--->浮点型  int c =32;  double d;  d = c;  printf("d=%f\n",d);  //3.double--->float 有可能发生精度的丢失  double e = 12.34567893;  float f;  f = e;  printf("f=%f\n",f);  //4.char 1Byte->int 4Byte  char c1 = 'd';  int num1;  num1 = c1;  printf("num1=%d\n",num1);  //5.int->char 有可能会发生数据错误  //512 256 128 64 32 16 8 4 2 1  // 1 0 0 1 0 0 0 0 1  //  int i = 289;  char c2;  c2 = i;  printf("c2=%d\n",c2);  return 0;  } |

标准化指数

100.3456=1.003456e2

0.00345=3.45e-3

printf()

|  |
| --- |
| #include<stdio.h>  int main(){  //1.d格式--->输出整型数据  printf("No.1:%d\n",122);  printf("No.2:%5d\n",122);//输出数据占5列（向右对齐）  printf("No.3:%-5d\n",122);//输出数据占5列（向左对齐）  //2.c格式--->单个字符  printf("No.4:%c\n",'\101');  //3.f格式--->浮点型数据（float,double）  printf("No.5:%f\n",3.14);//3.140000  float a = 10.0f;  printf("No.6:%20.15f\n",a/3);//占20列，15位小数  printf("No.7:%-20.15f\n",a/3);  //4.e格式--->结果为指数形式  printf("No.8:%25.15e\n",a/3);  printf("No.9:%e\n",190.34);  printf("No.10:%e\n",0.002345);  //5.o格式-- >将一个数按照8进制输出  printf("No.11:%o\n",1001);  //6.x格式-- >将一个数按照16进制输出  printf("No.12:%x\n",1001);  //7.u格式-->输出一个无符号的数  printf("No.13:%u\n",1001);  return 0;  } |

输入：scanf()

|  |
| --- |
| #include<stdio.h>  int main(){  int a;  scanf("a=%d",&a); //输入格式,往什么地方输入  //&-->取地址符  printf("a = %d\n",a);  printf("a(addr)=%p\n",&a);  printf("a(addr)=%d\n",&a);  return 0;  } |

另：

|  |
| --- |
| #include<stdio.h>  int main(){  float a,b,c;  scanf("a=%f,b=%f,c=%f",&a,&b,&c);  printf("a=%f,b=%f,c=%f",a,b,c);  return 0;  } |

特例1：输入数字时：

|  |
| --- |
| #include<stdio.h>  int main(){  int a,b,c;  scanf("%d%d,%d",&a,&b,&c);  printf("a=%d,b=%d,c=%d",a,b,c);  //1.每输入一个数据后，加空格  //2.每输入一个数据后，加回车  //3. 每输入一个数据后，加tab  return 0;  } |

特例2：输入字符时

|  |
| --- |
| #include<stdio.h>  int main(){  char a,b,c;  scanf("%c%c%c",&a,&b,&c);  printf("a=%c,b=%c,c=%c",a,b,c);  //对于字符输入时，每一个字符要紧紧的挨在一起。  return 0;  } |

**字符**输入输出函数：

1.输出：putchar();

|  |
| --- |
| #include<stdio.h>  int main(){  char a = 'a';  char b = 'r';  char c = 'w';  int d = 65;  putchar(a); //将变量a的值输出  putchar('\n');  putchar(b);  putchar(c);  putchar('\101');  putchar(d);  return 0;  } |

2.字符的输入：getchar()

调用一次getchar()函数，从键盘中得到一个字符。

|  |
| --- |
| #include<stdio.h>  int main(){  char a,b,c;  a = getchar();//获得到键盘输入的单个字符  b = getchar();  c = getchar();  putchar(a);  putchar(b);  putchar(c);  return 0;  } |

|  |
| --- |
| #include<stdio.h>  int main(){  int x,y;  scanf("%d%3d",&x,&y);  printf("%d %d",x,y);  return 0;  } |

例如：

|  |
| --- |
| #include<stdio.h>  #include<math.h>  int main(){  //求三角形面积  //海伦公式  double a,b,c;//三个边长  printf("请输出三个边的边长：\n");  scanf("a=%lf,b=%lf,c=%lf",&a,&b,&c);  double s,area;  s = (a+b+c)/2;  area = sqrt(s\*(s-a)\*(s-b)\*(s-c));//sqrt(3)  printf("三角形的面积是：%f",area);  return 0;  } |

求一元二次方程的根。

|  |
| --- |
| #include<stdio.h>  #include<math.h>  int main(){  float a,b,c;  printf("请输入a,b,c的值：\n");  scanf("%f,%f,%f",&a,&b,&c);  float p,q;  p = -b/(2.0\*a);  q = sqrt(b\*b-4\*a\*c)/2.0\*a;  float x1,x2;  x1 = p+q;  x2 = p-q;  printf("x1=%f,x2=%f",x1,x2);  return 0;  } |

第4章 选择结构

1.关系运算“>”“<”“==”“>=”“<=”“!=”

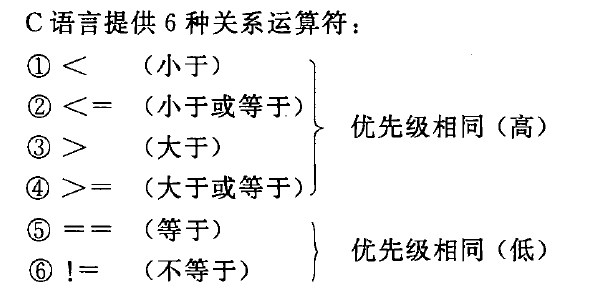
作用：判断左右两边的值**是否**满足。

3>2:判断3是否大于2，如果条件满足，返回值1。如果不满足，返回值0。

3>=3:返回值1。

优先级：

**算术运算符>关系运算符>赋值运算符**

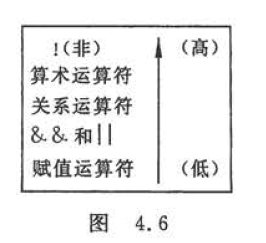


|  |
| --- |
| #include<stdio.h>  int main(){  int a =1,b=3,c=2;  printf("%d\n",(a==b>c));  //1.b>c 3>2=1  //2.a==1  printf("%d\n",(a<=b==c));  printf("%d\n",(a+1>b==c));  //1.a+1  //2.2>b-->0  //3.0==c  return 0;  } |

逻辑运算符 与(&&) 或(||) 非(!)

作用：判断逻辑运算符左右的表达式是否满足条件。满足：1.不满足：0

|  |
| --- |
| #include<stdio.h>  int main(){  printf("%d\n",(1&&1));  printf("%d\n",(1&&0));  printf("%d\n",(0&&1));  printf("%d\n",(0&&0));    printf("%d\n",(1||1));  printf("%d\n",(1||0));  printf("%d\n",(0||1));  printf("%d\n",(0||0));    printf("%d\n",(!0));  printf("%d\n",(!1));  return 0;  } |



|  |
| --- |
| #include<stdio.h>  int main(){  int a = 10;  int b = -2;  printf("%d",(a&&b));  //非0的数表示真,0表示假  return 0;  } |

|  |
| --- |
| #include<stdio.h>  int main(){  int a = 2,b=4,c=3,d=7;  printf("%d\n",a<b||c<d);//1||1--->1  printf("%d\n",a<b&&c<d);//1  printf("%d\n",a>b||c<d);//0||1--->1  printf("%d\n",a<b&&c>d);//1&&0--->0  printf("%d\n",!a<b&&c>d);//0<4&&c>d-->0  return 0;  } |

例题：

|  |
| --- |
| #include<stdio.h>  int main(){  int a = 1,b=9;  ++a||b++;//2||?=1  printf("b=%d",b);  return 0;  } |

a&&b，对于这个表达式，a如果确定为假，整个表达式为假，b这个表达式不执行。

a||b, 对于这个表达式，a如果确定为真，整个表达式为真，b这个表达式不执行。

Y = 5>3&&8<4-!0

= 5>3&&8<4-1

= 5>3&&8<3

= 1&&0=0

选择结构：if…else…

if(条件：条件表达式、逻辑表达式,){条件满足}else{条件不满足}

|  |
| --- |
| #include<stdio.h>  int main(){  //例如：输入两个整数，由大到小输出。  int a,b,t;  scanf("%d%d",&a,&b);//12 6 6 12  if(a<b){  t = a;  a = b;  b = t;  }  printf("%d,%d",a,b);  return 0;  } |

输入一个年份，判断此年份是否为闰年？year

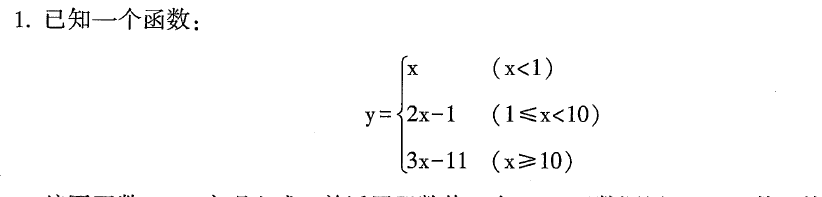
判断条件：1.能被4整除，但不能被100整除；2.能被400整除。

(year%4==0&&year%100!=0)||(year%400==0)

|  |
| --- |
| #include<stdio.h>  int main(){  int year;  printf("请输入年份：\n");  scanf("%d",&year);  if((year%4==0&&year%100!=0)||(year%400==0)){  printf("是闰年！");  }else{  printf("不是闰年！");  }  return 0;  } |

如果,在if..else语句中，不论是if中还是else中，只有1条语句，{}是可以省略的。

|  |
| --- |
| #include<stdio.h>  int main(){  int year;  printf("请输入年份：\n");  scanf("%d",&year);  if((year%4==0&&year%100!=0)||(year%400==0))  printf("是闰年！");  else  printf("不是闰年！");  return 0;  } |



输入x的值，求y。

|  |
| --- |
| #include<stdio.h>  int main(){  int x,y;  scanf("%d",&x);  if(x<1){  y = x;  }else if(1<=x&&x<10){  y=2\*x-1;  }else if(x>=10){  y = 3\*x-11;  }  printf("y = %d",y);  return 0;  } |

条件表达式（唯一一个三目运算）

例如：给定两个数，输出最大值。

表达式1?表达式2:表达式3;

执行步骤：

首先执行表达式1，如果表达1为真，执行表达式2；如果表达式1为假，执行表达式3。

|  |
| --- |
| #include<stdio.h>  //输入一个字符，判断它是否是大写字母，如果是大写字母  //转换为小写字母，如果是小写字母，直接输出。  int main(){  char ch,c;  ch=getchar();  c=(ch>='A'&&ch<='Z')?ch+32:ch;  putchar(c);  return 0;  } |

switch语句

switch(表达式){//整数类型(字符类型)

case 常量:语句;

case 常量:语句;

。。。。。

default:语句;//可以省略的

}

|  |
| --- |
| #include<stdio.h>  int main(){  //用户输入成绩的等级，显示用户的分数  //A-->85分以上,B->70-84,C->60-69,D->60以下  char grade;  scanf("%c",&grade);  switch(grade){  case 'A':printf("85分以上");//break跳出  case 'B':printf("70-84");break;  case 'C':printf("60-69");  case 'D':printf("60以下");break;  default:printf("输入错误！");  }  //当程序碰到一个匹配的case后 ，会从当前语句开始依次执行  //直到碰到break语句或switch语句执行完。  return 0;  } |

求解一元二次方程



1. 当a=0，不是二次方程
2. 当，有两个相等的解。
3. 当，有两个不相等的解。
4. 当，有两个复数根。

，



|  |
| --- |
| #include<stdio.h>  #include<math.h>  int main(){  double a,b,c,disc,x1,x2,p,q; //-0.00000001  printf("请输入abc的值\n");  scanf("%lf%lf%lf",&a,&b,&c);  if(fabs(a)<=1e-6){//a==0  printf("不是一元二次方程\n");  }else{  disc=b\*b-4\*a\*c;  if(fabs(disc)<=1e-6){//disc==0  printf("两个相等的实数根:%f\n",(-b/(2\*a)));  }else if(disc>1e-6){//1 2 2  x1 = (-b+sqrt(disc))/(2\*a);  x2 = (-b-sqrt(disc))/(2\*a);  printf("no1:%f,no2:%f",x1,x2);  }else{  p = -b/(2\*a);//实数  q = sqrt(-disc)/(2\*a);//虚数  printf("no1:%8.4f+%8.4fi\n",p,q);  printf("no2:%8.4f-%8.4fi\n",p,q);  }  }  return 0;  } |

1. 循环结构

while for

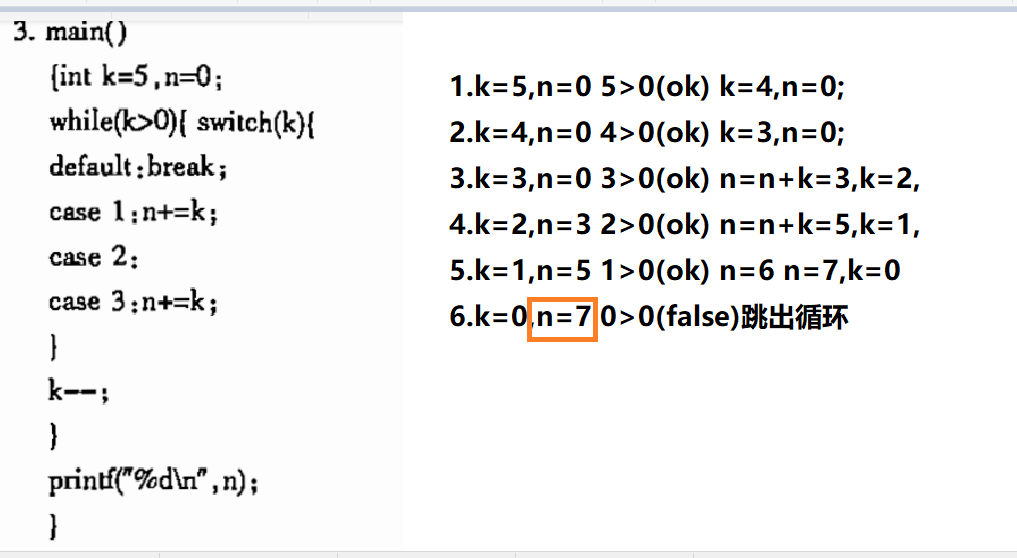
1. while语句

while(循环条件){循环体}

(1)初始值(2)循环体(3)循环增量

计算：1+2+3+..+100

|  |
| --- |
| #include<stdio.h>  int main(){  //1+2+3+...+100  int i = 1;  int sum = 0;  while(i<=100){  sum = sum+i;  i++;  }  printf("sum=%d\n",sum);  printf("i=%d\n",i);  //1.i=1 sum=0 i<=100(ok) sum =0+1=1 i=2  //2.i<=100(ok) sum = 1+2=3 i=3  //3.i<=100(ok) sum = 3+3=6 i=4  //4.i<=100(ok) sum = 6+4=10 i=5  //....  //100.i<=100(ok) sum =4950 +100=5050 i=101  //101.i<=100(false) 跳出循环  return 0;  } |



do{....}while();

|  |
| --- |
| #include<stdio.h>  int main(){  //1+2+3+...+100  int i =1;  int sum = 0;  do{  sum+=i;//sum = sum+i  i++;  }while(i<=100);  printf("sum=%d",sum);  return 0;  } |

while和do...while的区别?

while是先进行判断，后执行循环体。

do...while先执行循环体，后进行判断。

|  |
| --- |
| #include<stdio.h>  int main(){  int i = 1;  int a = 0;  while(i<1){  a = 100;  i++;  }  printf("a = %d",a);  return 0;  } |

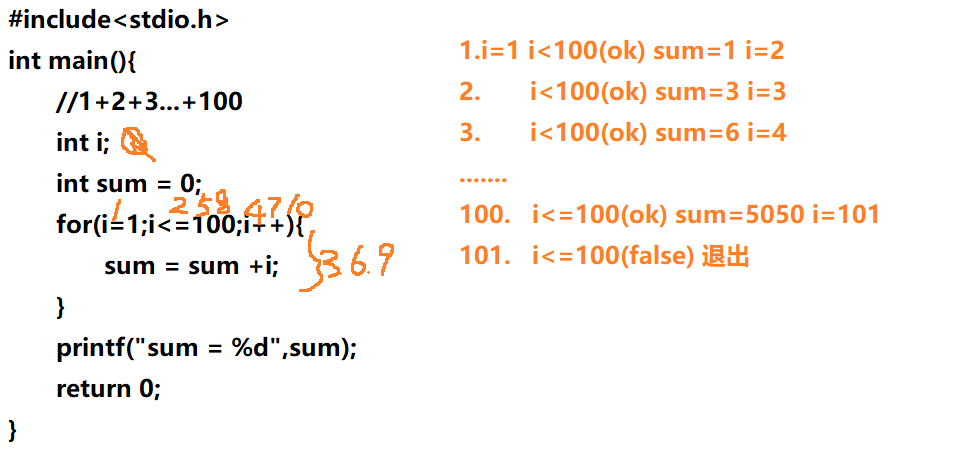
|  |
| --- |
| #include<stdio.h>  int main(){  int i = 1;  int a = 0;  do{  a = 100;  i++;  }  while(i<1);  printf("a = %d",a);  return 0;  } |

for循环

for(初值;循环条件;循环增量){循环体}

for循环和while循环是等价的。

|  |
| --- |
| #include<stdio.h>  int main(){  //1+2+3...+100  int i;  int sum = 0;  for(i=1;i<=100;i++){  sum = sum +i;  }  printf("sum = %d",sum);  return 0;  } |



for循环的几种形式？

1. 初值可以省略：for(;i<=100;i++)
2. 循环条件可以为空。for(i = 1;;i++)，为死循环
3. 循环增量也可以省略。for(i = 1;i<=100;)，为死循环。
4. 初始值,循环条件，循环增量都不写。for(;;)，为死循环，相当于while(-5){}
5. 初值表达式可以是多个表达式。

for(i = 1,sum=0;i<=100;i++)

1. 在c99中，初值处可以定义变量。

for(int i=1;i<=100;i++)

break关键字和continue关键字

break关键字：跳出整个循环

|  |
| --- |
| #include<stdio.h>  int main(){  for(int i=1;i<=10;i++){  if(i%5==0) break;  printf("%d\t",i);  }  return 0;  } |

continue关键字：跳出本次循环

|  |
| --- |
| #include<stdio.h>  int main(){  for(int i=1;i<=10;i++){  if(i%5==0) continue;  printf("%d\t",i);  }  //1.1  //2.2  //3.3  //4.4  //5.这一次循环不执行  //6. 6  //7.7  //8.8  //9.9  //10.这一次循环不执行  return 0;  } |

例题：输出100-200之间不能被3整除的数。

|  |
| --- |
| #include<stdio.h>  int main(){  for(int i=100;i<=200;i++){  if(i%3==0) continue;  printf("%d\n",i);  }  return 0;  } |

两层循环

输出一个4\*5的矩阵：

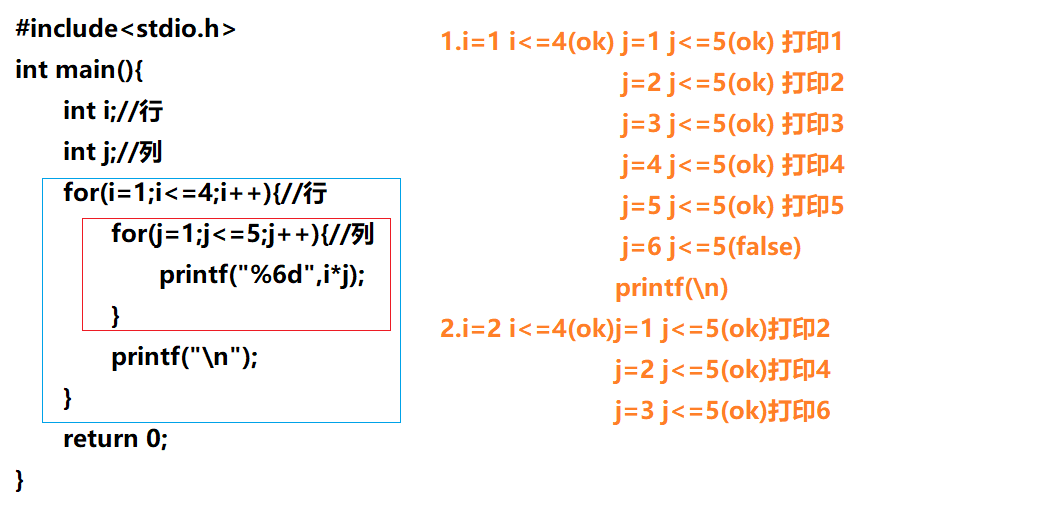
1 2 3 4 5

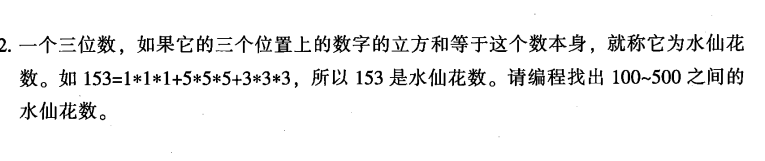
2 4 6 8 10

3 6 9 12 15

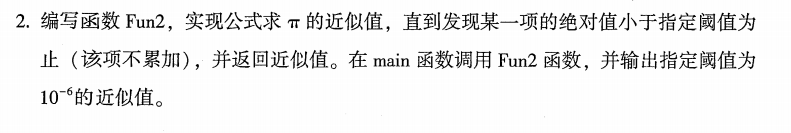
4 8 12 16 20

|  |
| --- |
| #include<stdio.h>  int main(){  int i;//行  int j;//列  for(i=1;i<=4;i++){//行  for(j=1;j<=5;j++){//列  printf("%6d",i\*j);  }  printf("\n");  }  return 0;  } |





|  |
| --- |
| #include<stdio.h>  int main(){//234/100=2 345/100=3 6789/1000  int b,s,g;//百 十 个  for(int i=100;i<=500;i++){  b=i/100;  s=i%100/10;  g=i%10;  if(i==b\*b\*b+s\*s\*s+g\*g\*g) printf("%d是水仙花数",i);  }  return 0;  } |





|  |
| --- |
| #include<stdio.h>  #include<math.h>  int main(){  int sigin = 1;// 分子  double n = 1 ;//分母  double pi =0.0; //pi值  double term = 1.0; //每一项  while(fabs(term)>=1e-8){  pi = pi +term;  n = n +2;  sigin = -sigin;  term=sigin/n;  }  pi = 4\*pi;  printf("%.18f",pi);  return 0;  } |

求：

|  |
| --- |
| #include<stdio.h>  int main(){  double s =0;  double t =1;  for(int i=1;i<=50;i++) {  t = t\*i;//t= 1 t=2 t=3\*2 t=6\*4  s = s +t;  }  printf("%22.15e",s);  return 0;  } |

数组

基本概念：有序数据的集合。

表示：数组名称和下标来表示数组，

a[3].

类型：同一数组中，每个元素的数据类型必须相同。

定义数组：

|  |
| --- |
| #include<stdio.h>  int main(){  int a[10];  //1.定义了一个整型int数组。在这个数组中，每一个元素  //都是int类型。  //2.a表示数组名称,表示数组首元素的地址,并且这个地址为常量  //3.10表示数组中有10个int类型的元素，数组中第一个元素的  //下标为0.  //4.会在内存中开辟连续的10个空间存放int类型的元素。  a[0] = 5; //0-->下标  a[1] = 10;  a[2] = 11;  a[6] = 0;  printf("%d\n",a);//a[0]元素的地址12345  printf("%d\n",a[0]);  printf("%d\n",a+1);//a[1]元素的地址12349  printf("%d\n",a+2);//a[2]元素的地址  for(int i=0;i<=9;i++){  printf("%d\t",a[i]);  }  printf("\n");  for(int i=0;i<=9;i++){  printf("%d\n",a+i);  }  return 0;  } |

一维数组的几种基本形式：

（1）对数组的每一个元素进行初始化

（2）对数组的部分元素进行初始化，没有赋值的元素，系统会自动补0。

（3）定义全“0”数组

（4）对数组的每一个元素初始化，省略数组的个数。

|  |
| --- |
| #include<stdio.h>  int main(){  //斐波那契数列fibonacci  //1 1 2 3 5 8 13...  //用数组求解fibonacci的前20项  int f[20]={1,1};  for(int i=2;i<=19;i++){  f[i] = f[i-2]+f[i-1];  }  for(int i=0;i<=19;i++){  if(i%5==0) printf("\n");  printf("%d\t",f[i]);  }  return 0;  } |

二维数组

a[5][5]，二维数组认为，每一个二维数组的元素是一个一维数组。

|  |
| --- |
| #include<stdio.h>  int main(){  int a[5][5];//行 列  //1.定义了一个整型二维数组,数组中的每一个元素都为  //整型int元素  //2.数组有5行5列，a表示首行元素的地址  //3.二维数组的行、列下标都是从0开始的。  //4.数组的每一个元素在内存中是按行顺序存放的  a[0][0] = 100;  printf("%d\n",a[0][0]);  printf("%d\n",a);//第一行元素的地址6487472  printf("%d\n",a+1);//第二行元素的地址6487492  printf("%d\n",a[0]);//第一个元素a[0][0]的地址  printf("%d\n",a[0]+1);//第二个元素a[0][1]的地址  printf("%d\n",a[1]+1);//第二行第二列a[1][1]的地址  return 0;  } |

二维数组赋值：

1.分行给二维数组赋值

2.将所有的数据写到一个花括号中。

3.对二维数组的部分元素赋值，没有赋值的元素为0。

4.全0二维数组。

5.对二维数组的全部元素进行赋值，对行可以省略，对列不能省略（在定义二维数组时）。

|  |
| --- |
| #include<stdio.h>  int main(){  //1.  int a[3][4] = {{1,2,3,4},{5,6,7,8},{9,0,1,2}};  printf("%d\n",a[2][1]);  for(int i=0;i<=2;i++){//行  for(int j=0;j<=3;j++){//列  printf("%d\t",a[i][j]);  }  printf("\n");  }  //2.  int b[3][4] = {1,2,3,4,5,6,7,8,9,0,1,2};  for(int i=0;i<=2;i++){//行  for(int j=0;j<=3;j++){//列  printf("%d\t",b[i][j]);  }  printf("\n");  }  //3.  int c[3][4]={{1,2},{2},{4}};  for(int i=0;i<=2;i++){//行  for(int j=0;j<=3;j++){//列  printf("%d\t",c[i][j]);  }  printf("\n");  }  //4.  int d[3][4]={0};  for(int i=0;i<=2;i++){//行  for(int j=0;j<=3;j++){//列  printf("%d\t",d[i][j]);  }  printf("\n");  }  //5.  int e[][4]={1,2,3,4,5,6,7,8,9,10,11,12};  //int e[3][]={1,2,3,4,5,6,7,8,9,10,11,12};错误！  //int e[][]={1,2,3,4,5,6,7,8,9,10,11,12};错误！  for(int i=0;i<=2;i++){  for(int j=0;j<=3;j++){  printf("%5d",e[i][j]);  }  printf("\n");  }    return 0;  } |



|  |
| --- |
| #include<stdio.h>  int main(){  int a[10];  //1.输入10个数  // for(int i=0;i<=9;i++){//&a[0]  // scanf("%d",&a[i]);  // }  for(int i=0;i<=9;i++){//a+1  scanf("%d",a+i);  }  //2.求平均分  float avg,sum = 0;  for(int i=0;i<=9;i++){  sum = sum +a[i];  }  avg = sum/10;  //3.统计比平均分少的个数  int j = 0;  for(int i=0;i<=9;i++){  if(avg>a[i]) j++;  }  printf("%d,%f",j,avg);  return 0;  } |

字符数组的定义和引用（字符串）

在c语言中将字符串存放到字符数组中。

1.形式一：

char a[10] = {'I',' ','l','o','v','e'};

2.形式二：

char b[]={"hello"};

**c语言规定用字符’\0’最为字符串结束的标志，当定义一个字符串时，系统会自动在这个字符后面加’\0’。**

**例如：在执行程序**

**printf(“hello world\n”);时，系统会在字符最后一个字符’\n’后加’\0’，系统会依次输出字符串的每一个字符，当遇到’\0’后就停止输出。**

3.形式三

char c[] = "hellohello";

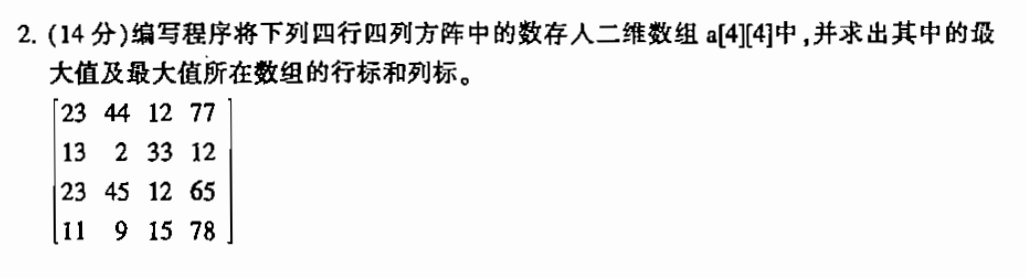
|  |
| --- |
| #include<stdio.h>  int main(){  char a[] = "helloworld!";//\0  printf("%s\n",a);  a[5] = '\0';  printf("%s",a);  // for(int i=0;a[i]!='\0';i++){  // printf("%c",a[i]);  // }  return 0;  } |

字符串的输入

|  |
| --- |
| #include<stdio.h>  int main(){  char str1[5],str2[5],str3[5];  //str1  scanf("%s%s%s",str1,str2,str3);  printf("str1=%s,str2=%s,str3=%s",str1,str2,str3);  //输入方式1：she he she  //输入方式2：she he she  //输入方式3：she回车he回车she  return 0;  } |

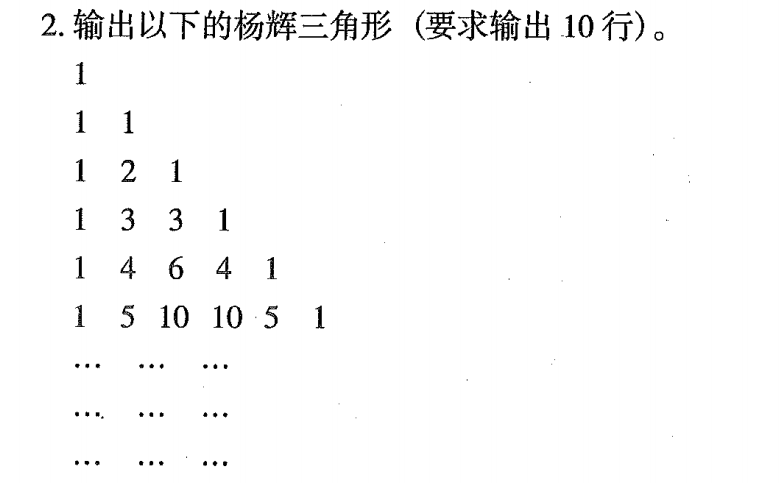
常用字符串操作函数

|  |
| --- |
| #include<stdio.h>  #include<string.h>  int main(){//string---字符串  char str1[] ="I love china!";  //1.输出字符串  puts(str1); //同printf("%s",str1);  //2.输入一个字符串  // char str2[10];  // gets(str2);  // puts(str2);  //3.字符串连接函数strcat(字符串1,字符串2);  //功能：将字符串2，拼接到字符串1中。  char str3[] = "hello";  char str4[] = "world";  strcat(str3,str4);//  puts(str3);  puts(str4);  //4.字符串的复制strcpy(字符串1,字符串2)  //功能：将字符串2复制到字符串1中。  char str5[10];  char str6[]="hello";  strcpy(str5,str6);  puts(str5);  //str5=str6;  //5.字符串比较函数strcmp(字符串1,字符串2)  //两个字符串相等，返回0;  //字符串1<字符串2,返回-1；  //字符串1>字符串2,返回1；  //1.按照字典的顺序比较大小的，越往后越大  //2.小写字母>大写字母 a>A  //3.如果出现不同的字符串，以第一个不同的字符比较为准  char str7[] ="helloa";  char str8[]="hellob";  printf("%d\n",strcmp(str7,str8));  //6.检查字符串的实际长度strlen()  char str9[] = "hello";//5  printf("%d\n",strlen(str9));//实际长度？  printf("%d\n",sizeof(str9));//占多大空间？  //7.将字符串转换为小写strlwr() lowercase  char str10[] ="BBBB";  strlwr(str10);  puts(str10);  //8.将字符串转换为大写 strupr() upercase  strupr(str10);  puts(str10);  return 0;  } |

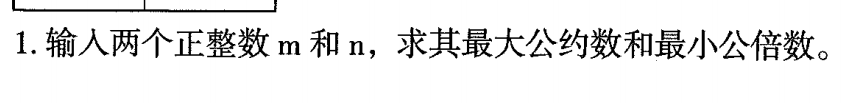


|  |
| --- |
| #include<stdio.h>  int main(){  //1.定义二维数组  int a[4][4] = {{23,44,12,77},{13,2,33,12}  ,{23,45,12,65},{11,9,15,78}};  //2.定义需要的变量  int max = a[0][0];//最大值  int x =0;//行标  int y =0;//列标  //3.算法---- 遍历  for(int i=0;i<=3;i++){  for(int j=0;j<=3;j++){  if(a[i][j]>max) max = a[i][j];  x = i;  y = j;  }  }  printf("max=%d,i=%d,j=%d",max,x,y);  return 0;  } |

|  |
| --- |
| #include<stdio.h>  int main(){  //1.定义二维数组  int a[4][4] = {{23,44,12,77},{13,2,33,12}  ,{23,45,12,65},{11,9,15,78}};  //2.定义需要的变量  int sum = 0;  //3.算法---- 遍历  for(int i=0;i<=3;i++){  for(int j=0;j<=3;j++){  sum = sum+a[i][j];  }  }  printf("sum=%d",sum);  return 0;  } |



|  |
| --- |
| #include<stdio.h>  int main(){  int a[10][10]={0};  for(int i=0;i<=9;i++){  a[i][0] = 1;  a[i][i] = 1;  }  for(int i=2;i<=9;i++){  for(int j=1;j<=9;j++){  a[i][j] = a[i-1][j-1]+a[i-1][j];  }  }  for(int i=0;i<=9;i++){  for(int j=0;j<=i;j++){  printf("%5d",a[i][j]);  }  printf("\n");  }  return 0;  } |



求:12和18的**最大公约数**。

12的约数：1 2 3 4 6 12

18的约数：1 2 3 6 9 18

求：3和4的**最小**公倍数。

3的倍数：3 6 9 **12** 15

4的倍数：4 8 **12** 16 20

最小公倍数=两个数相乘/**最大公约数**

|  |
| --- |
| **#include<stdio.h>**  **int main(){**  **printf("请输入两个值\n");**  **int m;**  **int n;**  **scanf("%d%d",&m,&n);**  **//1.找m和n最小的那个数**  **int temp;//保存最小的那个数**  **if(m>n) temp =n;**  **else temp =m;**  **//2.从最小值开始除，直到m和n第一次被同时整除，跳出循环**  **int i;**  **for(i = temp;i>0;i--){**  **if(m%i==0&&n%i==0){**  **printf("最大公约数为：%d\n",i); break;**  **}**  **}**  **int lcm;//**  **lcm = m \* n /i;**  **printf("最小公倍数为：%d\n",lcm);**  **return 0;**  **}** |

**例如：输入一个整数（>=3），判断这个数是否是为素数。**

**素数：只能被1和它本身整除的数。**

**5：1 5 7：1 7 8：1 8 2 4**

|  |
| --- |
| **#include<stdio.h>**  **int main(){**  **int n;//要判断的这个数**  **scanf("%d",&n);**  **int i;**  **for(i=2;i<=n-1;i++){**  **if(n%i==0) break;**  **}**  **//n=5**  **//i=2 2<=4 (ok) 5%2**  **//i=3 3<=4(ok) 5%3**  **//i=4 4<=4(ok) 5%4**  **//i=5**  **if(i<n) printf("不是素数\n");**  **else printf("是素数\n");**  **return 0;**  **}** |

**求100-200之间所有的素数。**

|  |
| --- |
| **#include<stdio.h>**  **int main(){**  **//求100-200之间所有的素数。**  **int flag = 0;//0表示素数 1表示不是素数**  **for(int i=100;i<=200;i++){//101**  **flag = 0;**  **for(int j = 2;j<=i-1;j++){**  **if(i%j==0){**  **flag= 1;**  **}**  **}**  **if(flag==0) printf("%5d",i);**  **}**  **return 0;**  **}** |

**第七章函数**

**函数function，表示完成一定功能的一个独立模块。**

**C程序是由1个或者多个模块组成的。**

**在C程序中多个函数都是相互独立的，一个函数并不属于另一个函数。**

**分类：库函数(stdio.h math.h string.h),用户自定义函数。**

**定义函数：**

**包括：名称，类型（返回值类型），参数，操作。**

**函数：无参函数和有参函数。**

**无参函数：**

**返回值 函数名称(){**

**函数的具体操作**

**}**

**有参函数：**

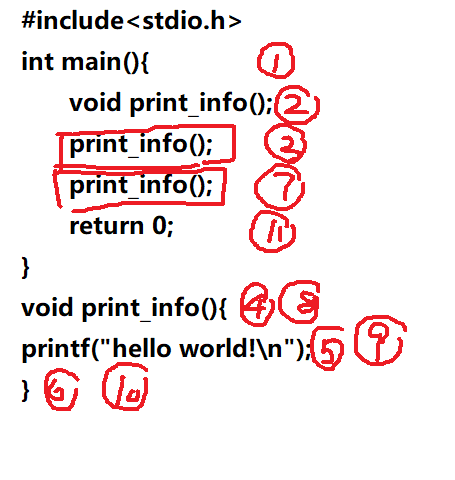
**返回值 函数名称(形式参数列表){**

**函数的具体操作**

**}**

|  |
| --- |
| **#include<stdio.h>**  **int main(){**  **//2.函数的声明--告诉编译器，定义了一个函数**  **void print\_info();**  **//3.使用函数（调用函数）**  **print\_info();**  **print\_info();**  **return 0;**  **}**  **//1.定义自定义函数**  **void print\_info(){//void返回值(空) print\_info函数名称**  **//()无参的函数**  **printf("hello world!\n");//函数的操作**  **}** |

|  |
| --- |
| **#include<stdio.h>**  **//1.定义自定义函数**  **void print\_info(){//void返回值(空) print\_info函数名称**  **//()无参的函数**  **printf("hello world!\n");//函数的操作**  **}**  **int main(){**  **//2.使用函数（调用函数）**  **print\_info();**  **print\_info();**  **return 0;**  **}** |



在函数调用时，实参向形参值传递的时候，是单向传递。

当调用函数时，系统会给函数的形参开辟临时的内存空间，当函数调用结束后，会立即释放形参在内存中的临时空间。

例如：输入两个整数，通过函数调用的形式，输出最大值。

|  |
| --- |
| #include<stdio.h>  int main(){  int max(int,int);  printf("请输入两个整数\n");  int a,b;  scanf("%d%d",&a,&b);  int s = max(a,b);  printf("s = %d",s);  return 0;  }  int max(int a,int b){  int z;  if(a>b){  z = a;  }else{  z = b;  }  return z;  } |

(1)x=2,y=3,z=0

(2)x=4,y=9,z=5

(3)x=2,y=3,z=0

|  |
| --- |
| #include<stdio.h>  int main(){  float max(float a,float b);  float a,b;  scanf("%f%f",&a,&b);  int s = max(a,b);  printf("%d",s);  return 0;  }  int max(float a,float b){  return a>b?a:b;  } |

函数的嵌套

函数不能嵌套定义，但可以嵌套调用。

|  |
| --- |
| #include<stdio.h>  int main(){  int max4(int,int,int,int);//1  int a=1,b=4,c=20,d=13;//2  int max = max4(a,b,c,d);//3  printf("max = %d",max);//19  return 0;//20  }  int max4(int a,int b,int c,int d){//4  int max2(int,int);//声明 5  int m = max2(a,b);//6  m = max2(m,c);//10  m = max2(m,d);//14  return m;//18  }  int max2(int a,int b){//7 11 15  if(a>b) return a;//8 12 16  else return b;//9 13 17  } |

函数的递归：

在调用一个函数的过程中，出现直接或间接调用该函数本身，称为递归。

循环===>递归

|  |
| --- |
| #include<stdio.h>  int main(){  int fun(int n);  fun(7);  ....  return 0;  }  int fun(int n){  ....  fun(n-1);  ....  return ...;  } |

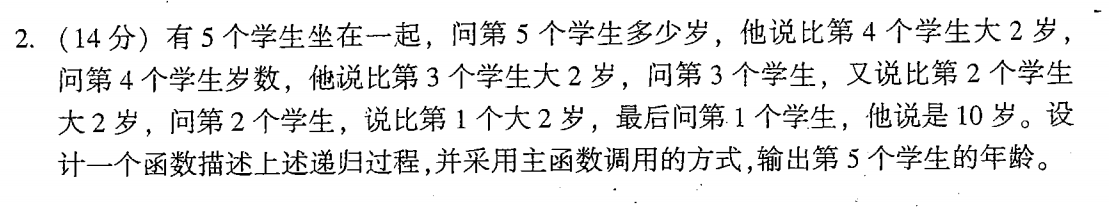
递归函数包含两个部分：1.函数的出口；2.递推公式（通项公式）

5！=5\*4\*3\*2\*1

|  |
| --- |
| #include<stdio.h>  int main(){  //5! = 5\*4\*3\*2\*1  int n = 1;  for(int i=1;i<=5;i++){  n = n \* i;  }  printf("%d",n);  return 0;  } |

5！

|  |
| --- |
| #include<stdio.h>  int main(){  int fun(int n);  int s = fun(5);  printf("%d",s);  return 0;  }  int fun(int n){  int f;  if(n==1||n==0){  f = 1;  }else if(n>=2){  f = fun(n-1)\*n;  }  return f;  } |



age(5) = age(4)+2

age(4) = age(3)+2

age(3) = age(2)+2

age(2) = age(1)+2

age(n) = age(n-1)+2

age(1) =10

|  |
| --- |
| //1+2+3+....+100  #include<stdio.h>  int main(){  int age(int n);  int age1 = age(5);  printf("age = %d",age1);  return 0;  }  int age(int n){  if(n==1) return 10;  else return age(n-1)+2;  } |

用递归函数实现程序。

|  |
| --- |
| #include<stdio.h>  int main(){  int apple(int n,int k);  int s = apple(3,5);  printf("%d",s);  return 0;  }  int apple(int n,int k){//表示n的k次方  if(k==0) return 1;  else return apple(n,k-1)\*n;  } |

用递归函数实现斐波那契数列。

数组作为函数的参数

1. 数组的元素作为函数的实参

**例如：有一个数组a[10],求出数组a中最大的那个数，和最大数对应的下标。**

|  |
| --- |
| **#include<stdio.h>**  **int main(){**  **int com(int,int);**  **int a[10] = {2,6,10,4,8,30,14,19,20,11};**  **int max = a[0];//a a+2 &a[4]**  **int i = 0;**  **for(int j=1;j<=9;j++){**  **if(com(max,a[j])>max) {**  **max = com(max,a[j]);**  **i = j;} }**  **printf("max = %d,index = %d",max,i);**  **return 0;**  **}**  **int com(int a,int b){**  **if(a>b) return a;**  **else return b;**  **}** |

（2）数组名称作为函数的形参

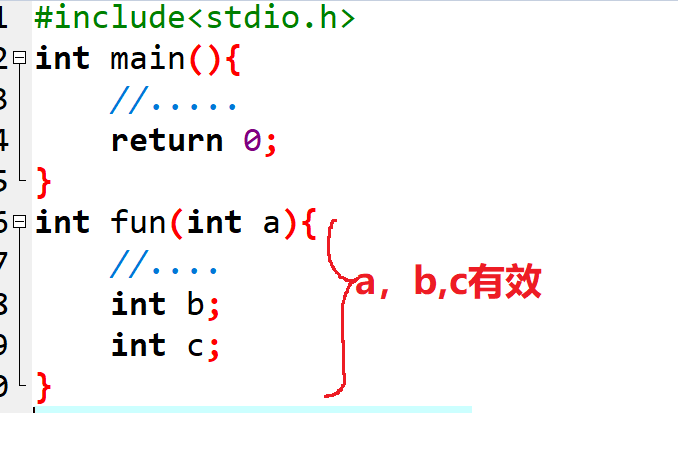
|  |
| --- |
| #include<stdio.h>  int main(){  void fun(int[]);  int a[5] ={1,2,3,4,5};  for(int i=0;i<=4;i++){  printf("%5d",a[i]);  }  printf("\n");  fun(&a[0]);//a  for(int i=0;i<=4;i++){  printf("%5d",a[i]);  }  return 0;  }//引用传递  void fun(int b[]){//int b[]--->只能接收地址  b[4] = 100;  } |

简单变量：int char long short double float....

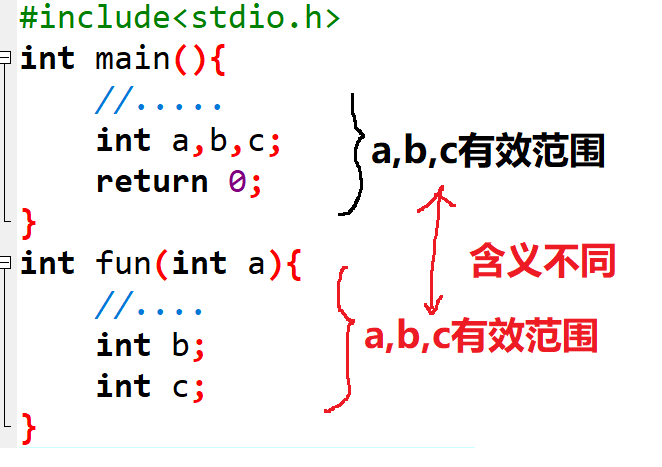
引用变量：数组[],指针\*

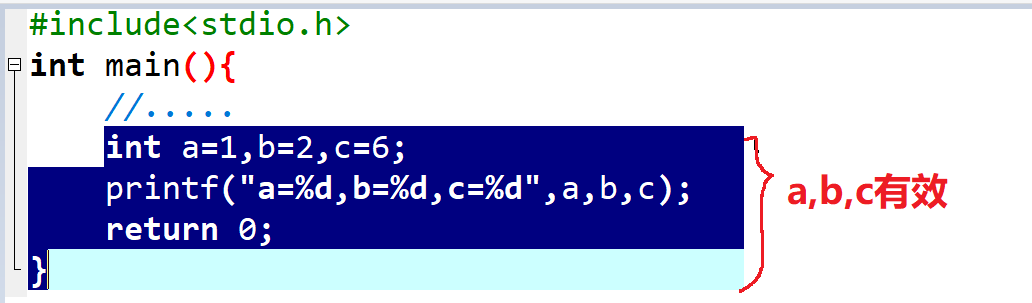
**局部变量**和**全局变量---变量的有效范围**

**局部变量：**



同一局部内，定义变量时，名称不能相同。

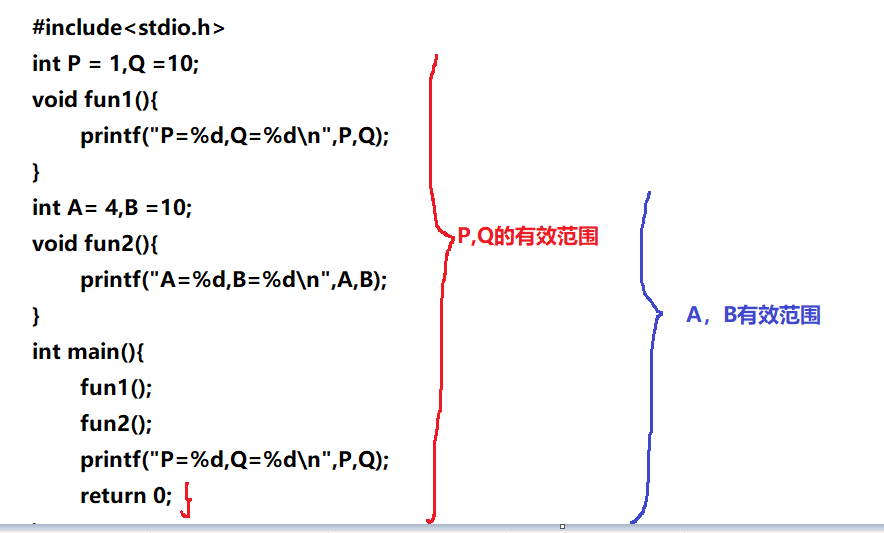




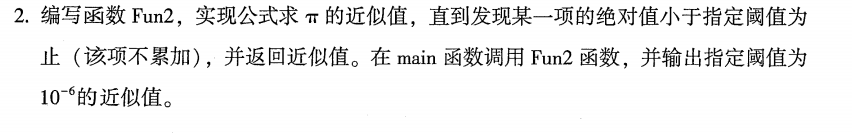
|  |
| --- |
| **#include<stdio.h>**  **int main(){**  **int i =10;**  **for(int i=0;i<=10;i++){**  **printf("%d\n",i);**  **}**  **printf("%d\n",i);**  **return 0;**  **}** |

|  |
| --- |
| **#include<stdio.h>**  **int main(){**  **int i =10;**  **for(i=0;i<=10;i++){**  **printf("%d\n",i);**  **}**  **printf("%d\n",i);**  **return 0;**  **}** |

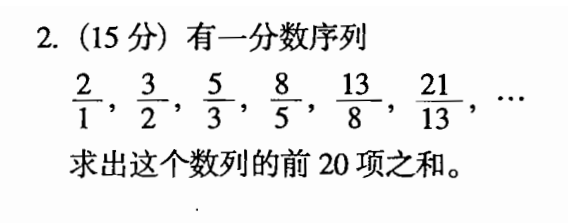
**全局变量：变量在“全局”有效。在定义全局变量时，通常使用大写字母。**



|  |
| --- |
| #include<stdio.h>  int P = 1,Q =10;  void fun1(){  int P=100,Q=100;  printf("P=%d,Q=%d\n",P,Q);  }  int A= 4,B =10;  void fun2(){  printf("A=%d,B=%d\n",A,B);  }  int main(){  fun1();  fun2();  printf("P=%d,Q=%d\n",P,Q);  return 0;  } |



|  |
| --- |
| #include<stdio.h>  #include<math.h>  int main(){  double Fun2(double error);  double s = Fun2(1e-6);  printf("%.16f",s);  return 0;  }  double Fun2(double error){//Fun2(1e-6)  int sign = 1;//分子  double n = 1.0;//分母  double term = 1.0;// 分子/分母  double pi = 0;  while(fabs(term)>=error) {  pi = pi +term;  sign = -sign;  n = n +2;  term = sign/n;  }  pi = 4\*pi;  return pi;  } |



|  |
| --- |
| #include<stdio.h>  int main(){  float a = 2;//分子  float b = 1;//分母  float sum = 0.0f;//和  for(int i=1;i<=20;i++){  sum = sum+a/b;  a = a+b;//新的分子  b = a-b;//新的分母  }  printf("%f",sum);  //1.sum = 0+2/1 a = 3 b =2  //2.sum = 0+2/1+3/2 a=5 b=3  //3.sum = 0+2/1+3/2+5/3  return 0;  } |

|  |
| --- |
| #include<stdio.h>  int main(){  float a[20];//分子  a[0] = 2;a[1] = 3;  for(int i=2;i<=19;i++){  a[i] = a[i-1]+a[i-2];  }  float b[20];//分母  b[0] = 1;b[1] = 2;  for(int i=2;i<=19;i++){  b[i] = b[i-1]+b[i-2];  }  float sum = 0;  for(int i=0;i<=19;i++){  sum = sum +a[i]/b[i];  }  printf("%f",sum);  return 0;  } |

；

变量的存储方式：**auto（自动变量），static（静态变量）**，register（寄存器变量），extern（外部变量）。

**auto（自动变量）：在函数中定义的变量，默认情况下，在函数调用时，会给定义的变量开辟临时的内存空间，当函数调用结束后，空间自动释放。**

|  |
| --- |
| **#include<stdio.h>**  **int main(){**  **int fun(int x);**  **printf("%d\n",fun(1));//12**  **printf("%d\n",fun(1));**  **return 0;**  **}**  **int fun(int x){**  **auto int a = 10;**  **a=a+1;**  **return (a+x);**  **}** |

**static（静态变量）：在函数中定义一个静态变量，在函数调用时，会给定义的变量开辟内存空间，当函数调用结束后，空间不释放。**

|  |
| --- |
| **#include<stdio.h>**  **int main(){**  **int fun(int x);**  **printf("%d\n",fun(1));//12**  **printf("%d\n",fun(1));**  **printf("%d\n",fun(1));**  **return 0;**  **}**  **int fun(int x){**  **static int a = 10;**  **a=a+1;**  **return (a+x);**  **}** |

1. 指针

指针：变量在内存中的地址。int a =100; &a表示变量a在内存中的地址。

|  |
| --- |
| #include<stdio.h>  int main(){  int a =100;  printf("a=%d\n",a);  printf("a的地址：%d\n",&a);  return 0;  } |

**直接访问**和**间接访问**。

**直接访问：直接从某个变量的地址空间中访问该变量的内容。**

**间接访问：某个变量i的地址存放到另一个变量j中，通过访问变量j找到变量i的地址，从而访问变量i的值。**

指针变量

如果有一个变量专门存放另一个变量的地址，称为指针变量。

定义指针变量：类型名称 \*指针变量名

例如：int \*a;int表示指针变量的类型；指针变量的类型用来表示指针变量可以”指向”变量的类型。a表示变量的地址；\*a表示该地址指向的值。

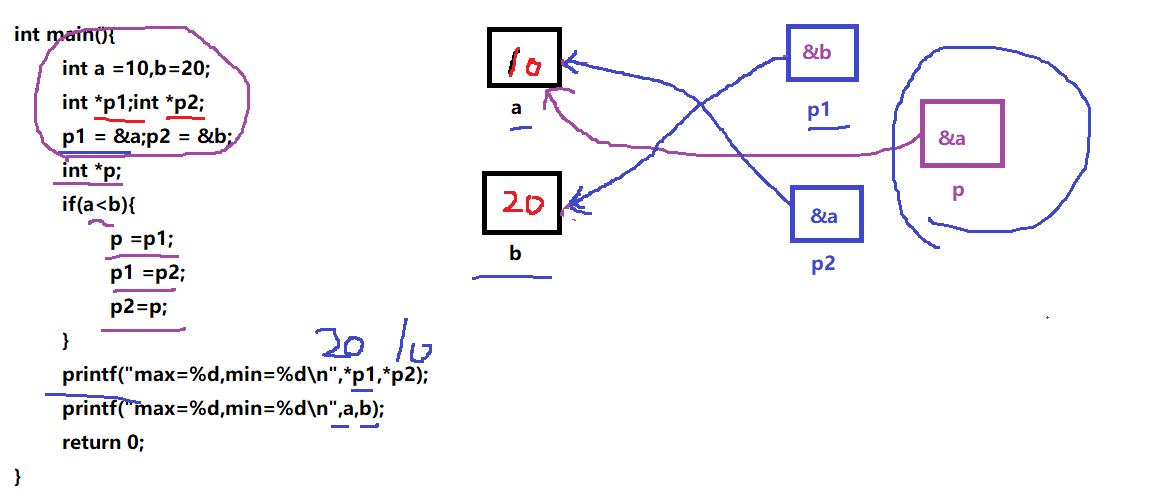
【注意】在定义指针变量时，必须指定指针变量的类型，什么类型的指针变量存放什么类型变量的地址，不能一会存放整型变量的地址，一会存放浮点数类型的地址。

|  |
| --- |
| #include<stdio.h>  int main(){  int a =100;  int \*p1;//定义了一个指针变量  p1 = &a;  printf("a = %d\n",a);  printf("a = %d\n",&a);  printf("a = %d\n",p1);  printf("a = %d\n",\*p1);  return 0;  } |

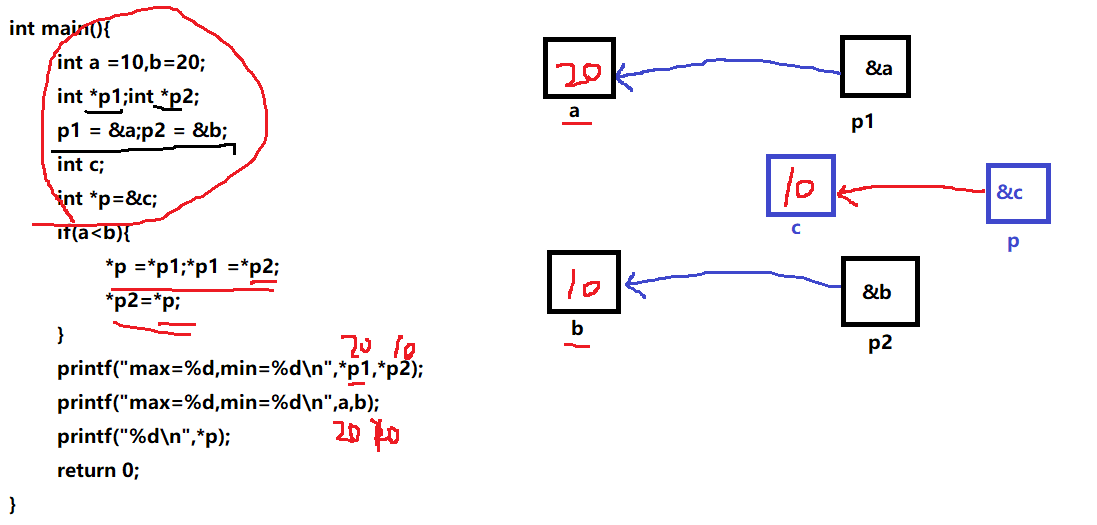
|  |
| --- |
| #include<stdio.h>  int main(){  int a =100;  int \*p1;//定义了一个指针变量  p1 = &a;  \*p1 = 88;  printf("a = %d\n",a);  printf("a = %d\n",&a);  printf("a = %d\n",p1);  printf("a = %d\n",\*p1);  return 0;  } |

|  |
| --- |
| #include<stdio.h>  int main(){  int a =100;  // int \*p1;//定义了一个指针变量  // p1 = &a;  int\* p1 = &a;  \*p1 = 88;  printf("a = %d\n",a);  printf("a = %d\n",&a);  printf("a = %d\n",p1);  printf("a = %d\n",\*p1);  return 0;  } |

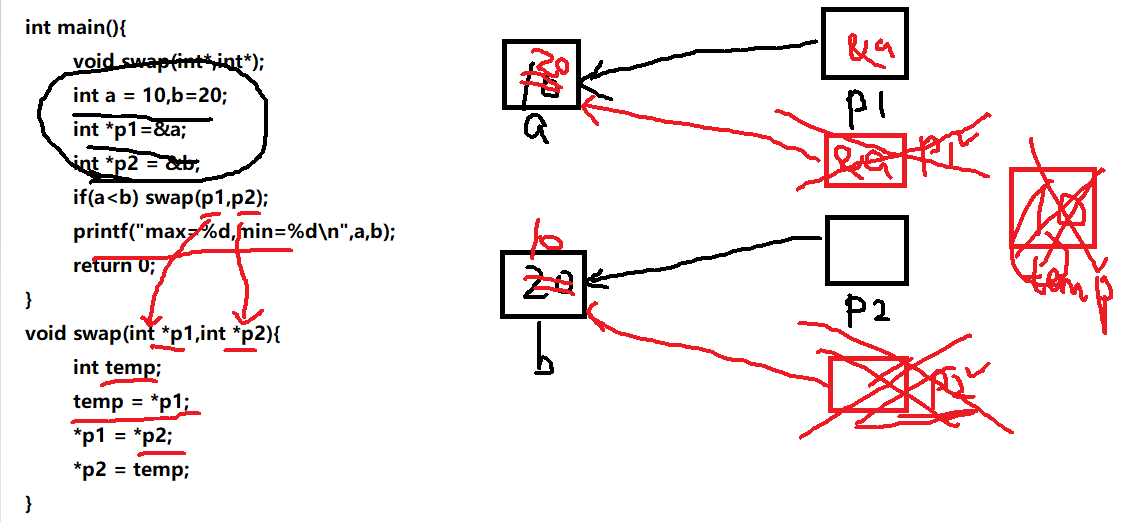
|  |
| --- |
| #include<stdio.h>  int main(){  int a =10,b=20;  int \*p1;int \*p2;  p1 = &a;p2 = &b;  int \*p;  if(a<b){  p =p1;  p1 =p2;  p2=p;  }  printf("max=%d,min=%d\n",\*p1,\*p2);  printf("max=%d,min=%d\n",a,b);  printf("%d\n",\*p);  return 0;  } |



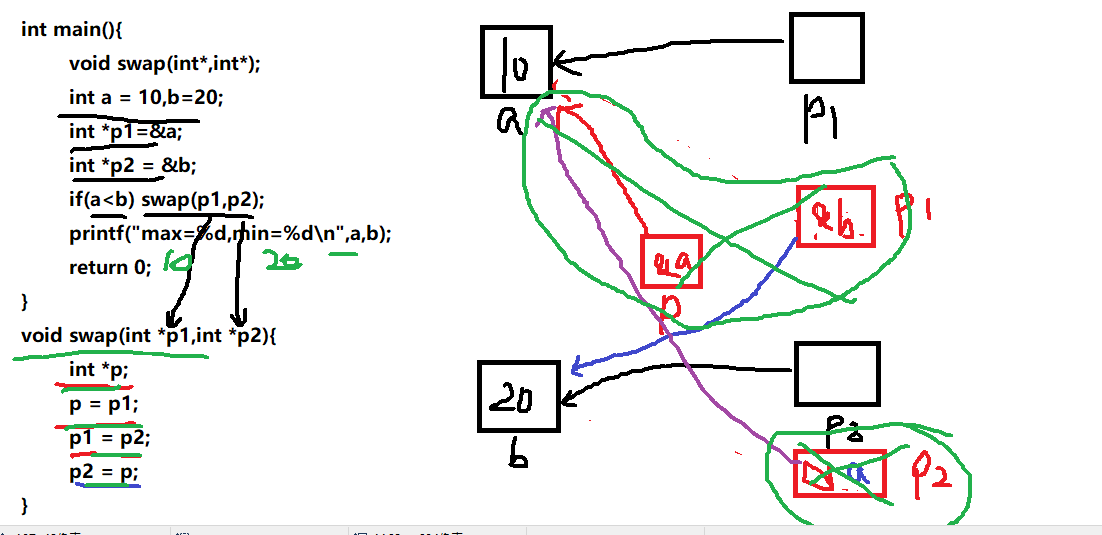
|  |
| --- |
| #include<stdio.h>  int main(){  int a =10,b=20;  int \*p1;int \*p2;  p1 = &a;p2 = &b;  int c;  int \*p=&c;  if(a<b){  \*p =\*p1;\*p1 =\*p2;  \*p2=\*p;  }  printf("max=%d,min=%d\n",\*p1,\*p2);  printf("max=%d,min=%d\n",a,b);  printf("%d\n",\*p);  return 0;  } |



指针变量作为函数形参



|  |
| --- |
| #include<stdio.h>  int main(){  void swap(int\*,int\*);  int a = 10,b=20;  int \*p1=&a;  int \*p2 = &b;  if(a<b) swap(p1,p2);  printf("max=%d,min=%d\n",a,b);  return 0;  }  //如果指针变量作为形参，实参向形参传递地址  void swap(int \*p1,int \*p2){  int temp;  temp = \*p1;  \*p1 = \*p2;  \*p2 = temp;  } |



|  |
| --- |
| #include<stdio.h>  int main(){  void swap(int\*,int\*);  int a = 10,b=20;  int \*p1=&a;  int \*p2 = &b;  if(a<b) swap(p1,p2);  printf("max=%d,min=%d\n",a,b);  return 0;  }  //如果指针变量作为形参，实参向形参传递地址  void swap(int \*p1,int \*p2){  int \*p;  p = p1;  p1 = p2;  p2 = p;  } |

通过指针引用数组

|  |
| --- |
| #include<stdio.h>  int main(){  int a[10] ={3,1,2,8,6,1,6,10,7,8};  //a a[0]//a+1 a[1]  int \*p;  p = a;  printf("%d\n",a);  printf("%d\n",p);  printf("%d\n",a+1);  printf("%d\n",p+1);  printf("%d\n",a[1]);  printf("%d\n",\*(p+1));  printf("%d\n",a[3]);  printf("%d\n",\*(p+3));  printf("%d\n",\*(a+3));  return 0;  } |

|  |
| --- |
| #include<stdio.h>  int main(){  int a[10] ={3,1,2,8,6,1,6,10,7,8};  for(int i=0;i<=9;i++){  printf("%d\t",a[i]);  }  printf("\n");  for(int i=0;i<=9;i++){  printf("%d\t",\*(a+i));  }  printf("\n");  int \*p = a;  for(int i=0;i<=9;i++){  printf("%d\t",\*(p+i));  }  return 0;  } |

总结：

int a[10] ={3,1,2,8,6,1,6,10,7,8};

int \*p;

p = a;

1. 指针变量指向数组a的第0个元素，也就是说指针变量p中保存的是数组a第0个元素的地址。
2. 如果指针p进行p=p+1运算，表示在同一个数组中指针p指向下一个元素。
3. p+i和a+i的含义相同，都表示数组元素a[i]的地址，即为&a[i].
4. \*(p+i)和\*(a+i)的含义相同，都表示数组元素a[i]的值，即为a[i].

|  |
| --- |
| #include<stdio.h>  int main(){  int a[5];  int \*p;  p = a;  for(int i=0;i<=4;i++){  scanf("%d",p);  p = p+1;//p++  }  p = a;  for(int i=0;i<=4;i++){  printf("%d\t",\*(p+i));  }  return 0;  } |

将数组a中n个整数按相反顺序存放到数组a中。

|  |
| --- |
| #include<stdio.h>  int main(){  void inv(int a[],int n);  int a[10] = {1,2,3,4,5,6,7,8,9,0};  int \*p = a;//int \*p = &a[0];  for(int i=0;i<=9;i++){  printf("%d\t",\*(p+i));  }  printf("\n");  inv(a,10);  for(int i=0;i<=9;i++){  printf("%d\t",\*(p+i));  }  return 0;  }  void inv(int a[],int n){  int i;//数组前半部分的下标  int j; //数组后半部分的下标  int temp;//临时空间  int m = n/2; //101/2=50交换次数  for(i=0;i<m;i++){  j=n-1-i;//9=10-1=n-1-0 8=n-1-1 7=n-1-2 6=n-1-3  temp = a[i];  a[i] = a[j];  a[j] = temp;  }  } |

|  |
| --- |
| #include<stdio.h>  int main(){  void inv(int \*a,int n);  int a[10] = {1,2,3,4,5,6,7,8,9,0};  int \*p = a;  for(int i=0;i<=9;i++){  printf("%d\t",\*(p+i));  }  printf("\n");  inv(a,10);  for(int i=0;i<=9;i++){  printf("%d\t",\*(p+i));  }  return 0;  }  void inv(int \*a,int n){  int \*i,\*j;  i = a;//指针i指向数组的第一个元素  j = a+n-1; //指针j指向数组的最后一个元素  int m = n/2;  for(;i<(a+m);i++,j--){  int temp = \*i;  \*i = \*j;  \*j = temp;  }  } |

通过指针引用字符串

字符串的引用方式：通过字符数组引用字符串和通过指针引用字符串。

|  |
| --- |
| #include<stdio.h>  int main(){  char str1[] = "helloworld";  //str1表示元素h的地址 \0结尾  printf("%s\n",str1);  //指针字符串  char \*str2 = "I love you!";  //str2表示字符串第一个元素I的地址，  //也就是说，指针str2指向元素I，字符串以\0结尾.  printf("%s\n",str2);  printf("%c\n",\*str2);  printf("%c\n",\*(str2+3));  char s[20] = "beijing";  char \*p;  p = s+1;  printf("%c\n",p[0]);  printf("%c\n",\*p);  printf("%c\n",\*(s+1));  printf("%c\n",s[1]);  return 0;  } |

例如：将字符串a复制给字符串b。

|  |
| --- |
| #include<stdio.h>  int main(){  char a[] = "student";//\0  char b[20];  char \*p1,\*p2;  p1 = a;  p2 = b;  for(;\*p1!='\0';p1++,p2++){  \*p2 = \*p1;  }  \*p2 = '\0';  printf("%s",b);  return 0;  } |

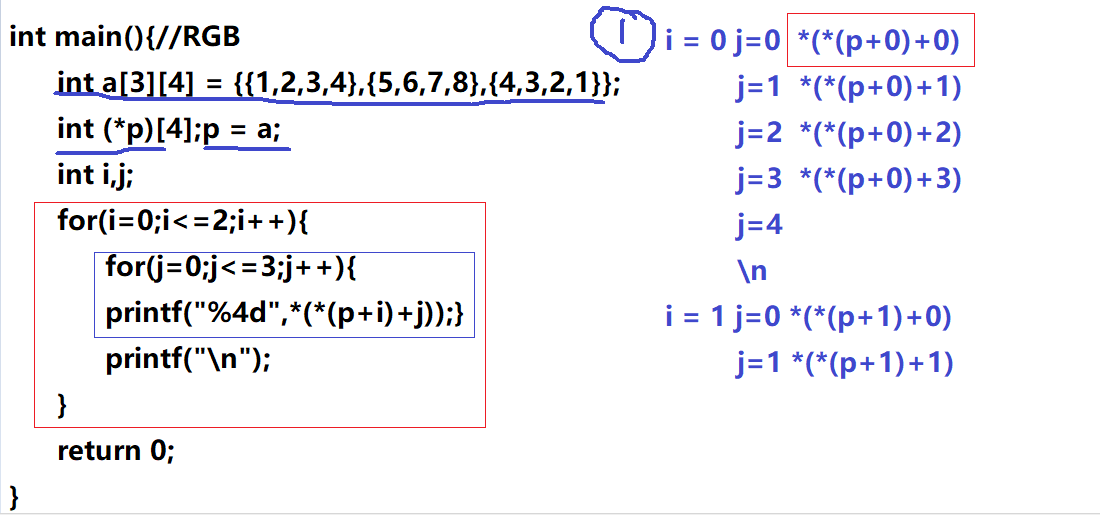
|  |
| --- |
| #include<stdio.h>  int main(){  void fun(char \*a,int \*b);  char \*a = "asdfgFGHJK45678";  int num = 0;  fun(a,&num);  printf("%d",num);  return 0;  }  void fun(char \*a,int \*b){  int i = 0; int n=0;  while(\*(a+i)!='\0'){  if(\*(a+i)>='a'&&\*(a+i)<='z') n++;  i++;  }  \*b = n;  } |

二维数组的指针表示

|  |
| --- |
| #include<stdio.h>  int main(){  int a[3][4] = {{1,2,3,4},{5,6,7,8},{4,3,2,1}};  printf("%d\n",a); //第0行元素的地址  printf("%d\n",a+1);//第1行元素的地址  printf("%d\n",&a[0][0]);//第0行第0列元素的地址  printf("%d\n",\*(a+0)+0);//第0行第0列元素的地址  printf("%d\n",\*(\*(a+0)+0));//第0行第0列元素的值  printf("%d\n",\*(\*(a+2)+2));//第2行第2列元素的值  printf("%d\n",\*(a+1)+2);//第1行第2列元素的地址  printf("%d\n",\*(\*(a+1)+2));//第1行第2列元素的值  printf("%d\n",\*a+1);//第0行第1列元素的地址  printf("%d\n",\*(\*a+1));//第0行第1列元素的值  printf("%d\n",a[1]+1);//第1行第1列元素的地址  printf("%d\n",\*(a[1]+1));//第1行第1列元素的值  printf("%d\n",a[1][1]);//第1行第1列元素的值  printf("%d\n",a[0]);//第0行第0列元素的地址  printf("%d\n",\*(a[0]));//第0行第0列元素的值  return 0;  } |

|  |
| --- |
| #include<stdio.h>  int main(){//RGB  int a[3][4] = {{1,2,3,4},{5,6,7,8},{4,3,2,1}};  int \*p;//指向一个元素  p = &a[0][0];// p = a[0];p = \*a  for(;p<a[0]+12;p++){//p = p+1  if((p-a[0])%4==0) printf("\n");  printf("%d\t",\*p);  }  return 0;  } |

|  |
| --- |
| #include<stdio.h>  int main(){//RGB  int a[3][4] = {{1,2,3,4},{5,6,7,8},{4,3,2,1}};  int (\*p)[4];//表示指向行的指针，这一行有4个元素。  p = a;  int i,j;  for(i=0;i<=2;i++){  for(j=0;j<=3;j++){  printf("%4d",\*(\*(p+i)+j));  }  printf("\n");  }  return 0;  } |



**指向函数**的指针

一般形式：类型名 (\*指针变量名)(函数的参数列表)

|  |
| --- |
| #include<stdio.h>  int main(){//RGB  int max(int,int);  int add(int,int);  int (\*p)(int,int);//可以指向一个函数的指针。  //返回值为int类型，并且有 2个参数，并且每个参数的  //类型为int的函数  p = max; //函数的名称表示，函数的地址  int z = (\*p)(7,8);  printf("%d\n",z);  p = add;  z = (\*p)(7,8);  printf("%d\n",z);  return 0;  }  int max(int x,int y){  // if(x>y)return x;  // else return y;  int z = (x>y)? x:y;  return z;  }  int add(int x,int y){  return (x+y);  } |

函数返回值为指针的函数：

一个返回值为指针（地址）的函数为返回值指针函数。

一般形式：类型名称 \*函数名(参数列表)

“stdlib.h”

void \*malloc(int size):在内存中动态分配size大小的空间。返回值为这个内存空间的首**地址**。

|  |
| --- |
| #include<stdio.h>  #include<stdlib.h>  int main(){  int \*p1;  //p1 = (int \*)malloc(20); //分配5个int类型大小的空间  p1 = (int \*)malloc(sizeof(int)\*5);  for(int i=0;i<=4;i++){  scanf("%d",p1+i);  }  for(int i=0;i<=4;i++){  printf("%d\n",\*(p1+i));  }  return 0;  } |

1. 结构体

结构体：是用户自己建立由不同类型数据组成的组合型数据结构。（**用户自定义的数据类型**）

回顾：整型(int short long)浮点型(flaot double)字符型(char) 数组 指针 结构体（**用户自定义的数据类型**）

一般形式：

1. struct 结构体名{ 成员列表}；
2. struct 结构体名{ 成员列表}变量名表；
3. struct { 成员列表}变量名表；

例如：定义一个学生结构体，学生类型包括学号，姓名，性别，住址。声明一个学生的变量，最后输出一个学生的信息。

|  |
| --- |
| #include<stdio.h>  #include<string.h>  struct Student {  int num;//学号  char name[20];//姓名  char sex;//性别  char addr[30];//地址  };  int main(){  //int i;  struct Student stu\_1;  stu\_1.num = 200101;  //strcpy()  strcpy(stu\_1.name,"zhangsan");  stu\_1.sex = 'F';  strcpy(stu\_1.addr,"taiyuan");  printf("%d %s %c %s\n",  stu\_1.num,stu\_1.name,stu\_1.sex,stu\_1.addr);  struct Student stu\_2;  stu\_2.num = 200102;  //strcpy()  strcpy(stu\_2.name,"lisi");  stu\_2.sex = 'M';  strcpy(stu\_2.addr,"beijing");  printf("%d %s %c %s",  stu\_2.num,stu\_2.name,stu\_2.sex,stu\_2.addr);  return 0;  } |

|  |
| --- |
| #include<stdio.h>  int main(){  struct Student {  int num;//学号  char name[20];//姓名  char sex;//性别  char addr[30];//地址  }stu\_1={201001,"zhangsan",'F',"beijing"};  printf("%d %s %c %s\n",  stu\_1.num,stu\_1.name,stu\_1.sex,stu\_1.addr);  return 0;  } |

|  |
| --- |
| #include<stdio.h>  int main(){  struct{  int num;//学号  char name[20];//姓名  char sex;//性别  char addr[30];//地址  }stu\_1={201001,"zhangsan",'F',"beijing"};  printf("%d %s %c %s\n",  stu\_1.num,stu\_1.name,stu\_1.sex,stu\_1.addr);  return 0;  } |

结构体数组：在数组中的每一个元素都是结构体变量。

|  |
| --- |
| #include<stdio.h>  struct Student{  int num;//学号  char name[20];//姓名  char sex;//性别  char addr[30];//地址  };  int main(){  struct Student stu[5]={  {200201,"zhangsan",'F',"taiyuan"},  {200202,"lisi",'F',"taiyuan"},  {200203,"wangwu",'M',"taiyuan"},  {200204,"zhaoliu",'F',"taiyuan"},  {200205,"sunqi",'M',"taiyuan"}  };  for(int i=0;i<=4;i++){  printf("%d %s %c %s\n",  stu[i].num,stu[i].name,stu[i].sex,stu[i].addr);  }    return 0;  } |

结构体指针

|  |
| --- |
| #include<stdio.h>  #include<string.h>  struct Student{  int num;//学号  char name[20];//姓名  char sex;//性别  char addr[30];//地址  };  int main(){  struct Student stu\_1;  struct Student \*p;  p = &stu\_1;  stu\_1.num = 200101;  strcpy(stu\_1.name,"zhangsan");  stu\_1.sex = 'F';  strcpy(stu\_1.addr,"taiyuan");  printf("%d %s %c %s\n",  stu\_1.num,stu\_1.name,stu\_1.sex,stu\_1.addr);  printf("%d %s %c %s\n",  (\*p).num,(\*p).name,(\*p).sex,(\*p).addr);  printf("%d %s %c %s\n",  p->num,p->name,p->sex,p->addr);  return 0;  } |

typedef声明新类型名称

|  |
| --- |
| #include<stdio.h>  #include<string.h>  typedef struct {  int num;//学号  char name[20];//姓名  char sex;//性别  char addr[30];//地址  }abc;  int main(){  abc stu\_1;  stu\_1.num = 200101;  strcpy(stu\_1.name,"zhangsan");  stu\_1.sex = 'F';  strcpy(stu\_1.addr,"taiyuan");  printf("%d %s %c %s\n",  stu\_1.num,stu\_1.name,stu\_1.sex,stu\_1.addr);  return 0;  } |