else配对原则：else不能单独存在，与if一起使用因此，与if或else if一起使用（上面，未使用的，离最近的）。

确定选择体的方法是：有括号看括号，没有括号，看第一个分号;

循环结构：重点

循环结构就是反复做有规律的事。

实现循环结构：C语言提供了几个关键字：

while do for 控制 break continue

1. while 循环

语法：

while(条件)

{循环体：反复做的有规律的事}

执行：先计算循环条件，如果结果为真，则执行循环体，然后继续计算循环条件，结果为真，执行循环提。直到循环条件结果为假，循环结束。

例题：

|  |
| --- |
| void main()  {int i=1;while(i<=3){printf(“test”);i++;}} //输出3次 test |

例题：

|  |
| --- |
| void main()  {int a=2,b=0,c=1,n=1;while(n<=3){c=c\*a; b=b+c; n++;}printf(“%d”,n,b);}//4 14 |

总结：做固定次数的循环结构：5次

int n=1; while(n<=5){n++;}

例题：一天3顿饭10天多少顿

|  |
| --- |
| void main()  {int i=1,sum=0;while(i<=10){sum=sum+3;i++;}printf(“%d”,sum);} |

例题：1-100求和

|  |
| --- |
| void main()  {int i=1,sum=0;while(i<=100){sum=sum+sum;i++;}printf(“%d”,sum);} //输出5050 |

例题：1!+2!+3+4...10! 阶层

|  |
| --- |
| void main()  {int i=1,sum=0,t=1; while(i<10){sum=sum+t; i++; t=t\*i;}} //阶层计算1 1x2 1x2x3 |

设计循环结构的第一种方法：定义变量法;

方法：先定义循环变量，给定初值，结合步进，利用循环条件，保证循环执行指定的次数。

过程：int i=1;while(i<=10){i++; 步进} 适合：循环次数是固定或已知的。

定义循环变量法，需要四要素：

初值：循环什么时候开始。

循环条件：控制循环到什么时候结束。循环结束条件与循环条件，二者互反。

循环体：循环反复做了什么。

步进：向结束条件靠近的程度。

例题：2/1+3/2+5/3+8/5….前20项和;

|  |
| --- |
| 分析：20项，循环20次，t单项：t=a/b 组合式单项  void main(){float i=1,a=2,b=1,k;float sum=0,t=a/b;while(i<=20) { sum=sum+t; i++; k=a; a=a+b; b=k; t=a/b; }printf(“%d”,sum);} |

类型题：数值累积型;多项式求和问题。

**解法**：

1. 定义单项，给定初值。

2. 循环中，单项做累加。

3. 为单项找规律。

2. do...while循环

语法：do{循环体}while(循环条件);

执行：先执行循环体，然后计算循环条件表达式。

例如：void main(){int x=3; do printf(“%d”,x-=2); while(--x);}

do while 和 while 的区别:前者体至少执行一次，后者体可能一次都不执行。

例题：求最大公约数（需要会算法）

|  |
| --- |
| void main()  {int m,n,r; scanf(“%d%d”,&m,&n); if(m<n){ r=m;m=n;n=r;} do{r=m%n;m=n;n=r;}while(r!=0);printf(“%d”,m);} |

3. for循环

语法：for(表达式1;表达式2;表达式3){循环体;}

for(初值;循环条件;步进){循环体;}

执行过程：初值→循环条件→循环体→步进→循环条件

例题：100以内有几个数同时能被3、5整除

|  |
| --- |
| {int n=0; for(int i=1;i<=100;i++){if(i%3==0&&i%5==0)n++;}printf(“%d”,n)} |

for循环的变形结构：3个表达式都可以省略（分号不能省略）

1.省略初值

说明：如果需要初值，需要放在循环的上方。

2.省略循环条件

说明：相当于循环条件结果恒为真，应该在循环体中使循环结束条件break

3.省略步进

说明：如果还需要步进，放到循环提中。

4.全部省略

break：彻底结束当前层的循环结构。

continue：结束当前层的本次循环，接着进行下一次的循环。

例题：

|  |
| --- |
| void main(){int n=0,s=0; for(;;;){if(n==5)break;n++;s+=n;}printf(“%d”,s)} |

三种循环结构的嵌套：就是在循环体中又使用了另一个循环结构，正常执行。

应用：1.输出二维平面图形，处理二位数组结构。

例题：99乘法表

|  |
| --- |
| void main(){  for(int i=1;i<=9;i++)  {for(int j=1;j<=i;j++){ printf(“%d\*%d=%d ”,i,j,i\*j); }printf(“\n”);}} |

例题：continue

|  |
| --- |
| 例如：void main(){int a=8,b=2;for(b=0;;b++){a++;if(a%2==0)continue;else b+=2; if(b>8)break;}printf(“%d%d”,a,b); //输出13 10} |

做循环结构要先确定体：有括号看括号，没有括号看地一个分号

例如：1+11+111+1111+11111 的和。

属于：数值累加型

编程步骤：

1. 定义单项和处值。t=1

2. 循环中单项做累加。sum=sum+t; 构造5次的循环

3. 为单项找规律：难点：后一项如何根据前一项计算出。t=t\*10+1

|  |
| --- |
| void main(){int i=1; int t=1,sum=0; while(i<=5){sum+=t;i++;t=t\*10+1}} |

设计循环的第一种方法：定义循环变量法，计算循环次数固定的情况。

方法：定义循环变量，设定初值，结合步进，利用循环条件，保证循环指向固定的次数。

设计循环的第二种方法：定义控制变量法，计算循环次数未知或不固定的情况。

方法：先设计无限次的循环，然后定义或寻找控制变量，当变量满足一定的条件，使循环结束（一般用break）

格式：while(1){if(k>0)break;}

例题：1+2+3+….+n 多少次的和大于等于5050。

|  |
| --- |
| void main(){int t=1,sum=0; while(1){ sum+=t; if(sum>=5050)break; t+=1;}printf(“%d”,t); // 输出100} |

例题：1-1/3+1/5-1/7+…..直到某项的绝对值相遇1e-6为止。（算法）

|  |
| --- |
| void main(){  float b=1,sum=0,t;t=1/b;  while(1){sum+=s\*t; if(fabs(t)<1e-6) break;s=s\*-1;b=b+2;t=1/b;}printf(“%f”,sum);} |

**函数**：

|  |
| --- |
| void main(){int a=3,b=4,k; if(a>b) k=a; else k=b;printf(“%d”,k); } |

修改：

|  |
| --- |
| int max(int a,int b){ int k; if(a>b) k=a; else k=b; return k;}  void main(){int a=3,b=4,k; k=max(a,b); printf(“%d”,k);} |

程序结构上的设计问题，设计程序时，应该将任务划分为多个小任务，然后找不同的函数帮忙完成。

学习函数要学会3方面内容：

定义函数 使用函数 声明函数

1.定义函数：就是根据预想的输入，输出和处理过程，写出完整的函数结构。

完整的函数结构包括：函数头+函数体。

函数头：返回值类型 函数名(形参) 形参完成了数据的输入

函数体：就是大括号扩起来的多条语句 ，完成处理过程和输出。

函数名：属于标识符，最好见名知意。

返回值类型：由函数计算后的结果类型决定的，但返回值的类型是否写出分两种情况：

(1)如果函数计算后产生的结果，需要交给调用者，则必须标出返回值类型，此时需要配套使用return关键字。int add(int a,int b){int k; k=a+b; return k;}

(2)如果函数计算后产生的结果，不需要交给调用者，则就不用标出返回值类型，

此时void代替类型。void add(int a,int b){int k; k=a+b;}

形参：是外界向函数内部传递已知信息的入口，有几个已知信息，就应该有几个形参。行参一般都是各种类型的变量。

说明：如果计算结果类型与返回值类型不同，以返回类型为准。

函数体：体内写出程序的处理过程，写算法：就是解决问题的方法和步骤

2.函数的调用：函数定义后，如果没被调用，则不起任何作用。

调用函数时，需要写出函数名的必要的实参，调用时分为两种情况。

(1)返回值类型为非空，调用函数时，需要找人接受返回值。int k=add(2,3);

(2)返回值类型为空，调用函数时，不需要找人接受返回值。add(2,3);

例题：int k=fun(‘a’,2.5); 请问函数首部形式为。（横线为实参）

|  |
| --- |
| int fun(char a,double b) |

函数调用时需要进行函数的参数传递：实参传递给形参

传递规则1：如果函数的形参是基本数据类型（包括整型、字符、浮点）变量，则实参传递给形参传递的是实参的拷贝值。实参与形参之间互不影响。

return 关键字的作用：

(1)与函数的返回值类型配套使用，返回结果值，但最多返回一个值。

(2)能结束函数的执行。

总结：函数定义后，不去调用，是不起作用的，调用时，需要函数名和必要的实参，实参与形参是一一对应的。

3. 函数的声明：

是解决函数的调用行位置与定义行位置，二者之间位置关系不匹配的问题。

如果函数的调用行在上，定义行位置在下，则二者不匹配，此时需要进行函数的声明。

void main(){….} int max(int a,int b){…..}

在函数的调用行上方，进行函数的声明：使用函数的原型进行声明;

函数的原型就是函数首部（头）后面加一个分号：

即：int max(int a,int b) void main(){….} int max(int a,int b){…..}

4. 函数的嵌套调用：就是在函数体内又调用了另一个函数

例如：

|  |
| --- |
| int fun1(int x){int y; y=x+x; return y;}  int fun2(int a,int b){int z; z=fun1(a\*b); return z;}  void main(){int a=2,b=5,c; c=fun2(a,b); printf(“%d”,c);} |

说明：函数可以嵌套调用，但不能嵌套定义。

5. 函数的递归调用：递归调用属于嵌套调用的一种特殊格式，就是在函数体内又调用了另一个函数，这个函数是自己。例如:void add(){add();}

例题：

|  |
| --- |
| int fun(int n){int k; if(n==1)k=1;else k=n\*fun(n-1);return k;}  void main(){int a=4,k; k=fun(a); printf(“%d”,k);} |

总结：递归调用：

1.函数所解决的问题必须能划分多个子问题,每次调用就相当于解决一个子问题。

2.函数体内必须有使函数结束的条件，否则递归就将变成死循环调用。如果函数包含参数，则参数一定要向结束条件方向靠近。

局部变量和全局变量：是从变量的使用范围角度，来分析。

局部变量：定义在函数内部，函数形参和代码段内部的变量。

使用范围：从定义行开始到本函数接触，或本代码段结束。

特点：系统不给默认初始值，需要人为赋值，局部变量被存储在内存的动态区。

局部变量用一次，被初始化一次。

全局变量：定义在函数外部的变量。

使用范围：从定义行开始，到本文件或其他文件结束。

特点:：系统给默认初始值0; 全局变量被存储在内存的静态区，具有保存上一次结果值的功能。全局变量仅被初始化一次（编译期间）。

全局变量与局部变量同名。则在局部变量的使用范围内，全局变化会被隐藏，起作用的是局部变量。