目录

[1.程序设计基础 1](#_Toc130304002)

[2.顺序结构程序设计 3](#_Toc130304003)

[3.选择结构程序设计 6](#_Toc130304004)

[4.循环结构程序设计 8](#_Toc130304005)

[6.模块化程序设计 14](#_Toc130304006)

# 1.程序设计基础

1. 了解程序设计语言的发展及特点；

1963年剑桥大学将ALG 60 发展成CPL语言

1967剑大的（马丁理查德）对CPL语言简化，产生BCPL语言

1970美国贝尔实验室（肯汤姆森）将BCPL精华提炼 B语言

1973美国贝尔实验室（丹尼斯瑞切）在b语言基础上设计出新语言C语言

1977娜娜 发表《可移植的C语言编译程序》

1978年（布瑞恩科尼森）和DennisM。Ritchie出版c

1983年–美国ANSL c标本被采用

特点：1.每条语句后面都以“；”作为终止符，它是C语句结束符。

1. 每一个程序必须有一个且只能有一个主函数，程序从主函数开始执行。
2. 在C语言中，大写小写字母是有区别的
3. 程序中可以加注释部分，有块注释和行注释两种方式
4. 了解程序设计语言源程序书写格式；
5. 程序中每行可写一条语句，也可以写多条语句，一般一行写一条语句
6. 程序书写要注意适当缩进，使程序清晰易读；
7. 程序花括号必须成对出现
8. 写程序时要习惯使用注释
9. 掌握程序设计语言源程序的基本结构；

注释（说明文字）[预处理](https://so.csdn.net/so/search?q=%E9%A2%84%E5%A4%84%E7%90%86&spm=1001.2101.3001.7020" \t "_blank)指令、主函数入口、主函数体

编译预处理行

全局变量声明

Void main(){

局部变量声明

程序段

}

返回数据类型f1（形式参数说明）{

局部变量声明

程序段

}

返回数据类型f2（形式参数说明）{

局部变量声明

程序段

}

…

返回数据类型fn（形式参数说明）{

局部变量声明

程序段

}

1. 掌握程序设计语言各种数据类型及其定义方法；
2. 整型 十进制数2，-983，八进制数以0开头，由0-7构成，012=12；

十六进制以0x开头，由0-9和a-f构成，0x12A，12A=298

1. 实型 float，double，long double
2. 字符型 A x %
3. 掌握程序设计语言各种运算符和表达式的应用。
4. 常量表达式

a,12,12.4

值：就是变量或者常量本身的值；

作为条件的时候非0即真

1. 算数表达式

a+b，c\*d+a，12/3+d，i++，--a

值：就是计算的结果

作为条件的时候非0即真

1. 赋值表达式

a=12，a+=12，a=c=d

值：赋值完a的值，就是结果

1. 作为条件的时候非0即真
2. 关系表达式

a>b，2==3

值：1和0（当表达式成立时返回一个1，表达式不成立返回一个0）

作为条件的时候非0即真

1. 逻辑表达式

a&&b(a与b)，c||b(c或b)，!a（非a）

值：1和0当表达式成立时返回一个1，表达式不成立返回一个0）

作为条件的时候非0即真

1. 复合表达式

x=（y=（a+b+a>4）,z=10）

值：根据运算符优先级和结合性

作为条件的时候非0即真

1. 逗号表达式

（1,2,3,4，a）

值：最右侧的值是逗号表达式的值

作为条件的时候非0即真

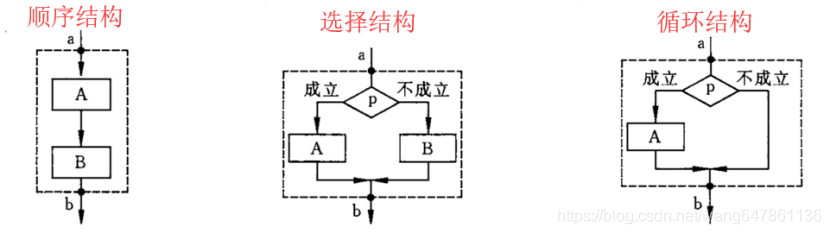
# 2.顺序结构程序设计

（1）了解程序和算法的概念；

程序是为实现特定目标或解决特定问题而用计算机语言编写的命令序列的集合。

算法是指为解决某个具体问题而采用的方法和步骤

（2）了解结构化程序设计的基本结构；



（3）了解流程控制的基本语句；

If-else,条件语句；switch-case，多分支选择语句；while，循环语句；do-while循环语句；foe，循环语句；break，终止循环或switch结构语句；continue，结束本次循环语句；goto转向语句；return返回语句；

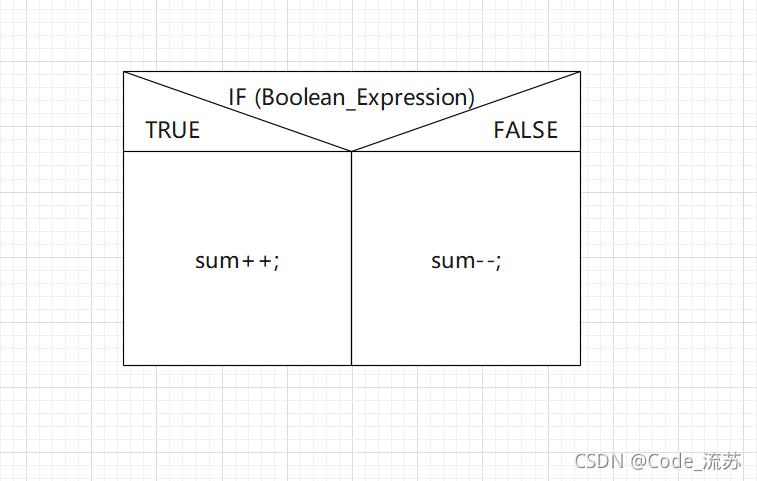
4）熟悉流程图和 N-S 图绘制；

N-S图

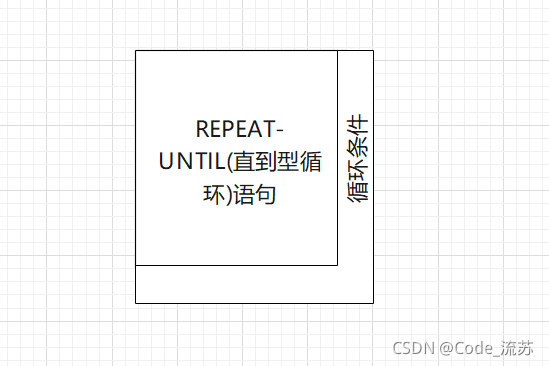
顺序结构

|  |
| --- |
| A |
| B |
| C |

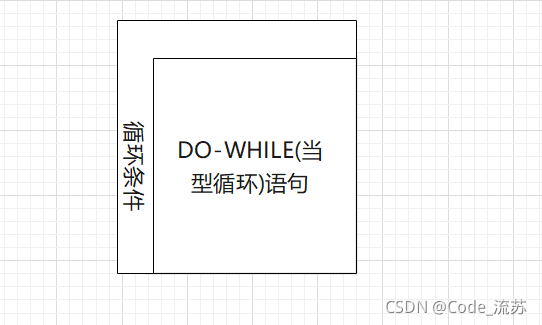
分支结构



直到循环



**当型循环**



（5）掌握字符输入/输出函数应用；

getchar：输入单个字符，保存到字符变量中；

gets：输入一行数据，保存到字符变量中；

putchar：输出单个字符；

puts：输出字符串；

（6）掌握格式输入/输出函数应用。

scanf：格式化输入函数，一次可以输入多个函数，保存到多个变量中；

printf：格式化输出函数，可以输出常量、变量；

printf：整数%d，字符%c，浮点数%lf，字符串%s

# 3.选择结构程序设计

（1）掌握关系表达式和逻辑表达式的定义及应用；

1. 关系表达式

a>b，2==3

值：1和0（当表达式成立时返回一个1，表达式不成立返回一个0）

作为条件的时候非0即真

1. 逻辑表达式

a&&b(a与b)，c||b(c或b)，!a（非a）

值：1和0当表达式成立时返回一个1，表达式不成立返回一个0）

作为条件的时候非0即真

（2）掌握用 if 语句实现单分支和双分支选择结构；

if(表达式) if (表达式)

语句A 语句A

else

语句B

（3）掌握 switch 语句实现多分支选择结构；

switch（表达式）

{

case 常量1 ： 语句1

case 常量2 ： 语句2

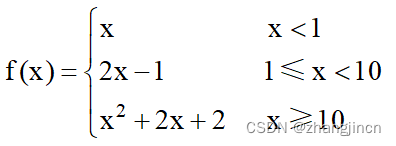
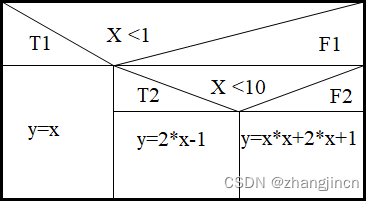
......

case 常量n : 语句n

default : 语句n+1

}

（4）了解简单的嵌套选择结构程序设计。

| **判断结果** | **满足条件** |
| --- | --- |
| T1 | X<1 |
| F1 | X>=1 |
| T2 | 1<=X<10 |
| F2 | X>=10 |

# 4.循环结构程序设计

（1）熟悉循环结构的构成和特点；

构成：初始化、判断、循环体、迭代

特点：在给定条件成立时，反复执行每段程序，直到条件不成立为止（一次又一次的执行相同代码）；

（2）理解循环控制变量的应用；

在这些语句中，循环体被重复执行的次数由循环条件控制，称为控制表达式（controlling expression）

（3）掌握用 for 语句实现循环结构；

#include <stdio.h>

int main(){

int i, sum=0;

for(i=1/\*语句①\*/; i<=100/\*语句②\*/; i++/\*语句③\*/){

sum+=i;

}

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

return 0;

}

（4）掌握用 while 和 do…while 语句实现循环结构；

#include <stdio.h>

int main(){

int i=1, sum=0;

while(i<=100){

sum+=i;

i++;

}

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

return 0;

}

#include <stdio.h>

int main(){

int i=1, sum=0;

do{

sum+=i;

i++;

}while(i<=100);

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

return 0;

}

（5）掌握 continue 和 break 语句的应用；

当 break 关键字用于 while、[for 循环](http://c.biancheng.net/view/172.html" \t "_blank)时，会终止循环而执行整个循环语句后面的代码。break 关键字通常和 if 语句一起使用，即满足条件时便跳出循环。

#include <stdio.h>

int main(){

int i=1, sum=0;

while(1){ //循环条件为死循环

sum+=i;

i++;

if(i>100) break;

}

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

return 0;

}

continue 语句的作用是跳过循环体中剩余的语句而强制进入下一次循环。continue语句只用在 while、for 循环中，常与 if 条件语句一起使用，判断条件是否成立。

#include <stdio.h>

int main(){

char c = 0;

while(c!='\n'){ //回车键结束循环

c=getchar();

if(c=='4' || c=='5'){ //按下的是数字键4或5

continue; //跳过当次循环，进入下次循环

}

putchar(c);

}

return 0;

}

（6）掌握二重循环结构的构建；

do {

//循环操作1

do {

//循环操作2

}while(循环条件1);

}while(循环条件2);

（7）了解 goto 语句的应用。

C 语言中的 **goto** 语句允许把控制无条件转移到同一函数内的被标记的语句。

#include<stdio.h>

#include<stdlib.h>

int main()

{

int i = 1;

while(1)

{

printf("在while（1）里\n");

while(i++)

{

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

if(i > 3)

{

goto TiaoChu;

}

}

}

TiaoChu:

printf("程序结束\n");

return 0;

}

运行结果：

标号位置

在while（1）里

2

3

4

程序结束  
5.数组应用

1. 掌握一维数组的定义及应用；

[一维数组](https://so.csdn.net/so/search?q=%E4%B8%80%E7%BB%B4%E6%95%B0%E7%BB%84&spm=1001.2101.3001.7020)指的是只有一个下标的数组，它用来表示一组具有相同类型的数据

类型说明符 数组名[常量表达式]; int array[5l;

1. 掌握字符数组的定义及应用；

**用来存放字符量的数组称为字符数组**    char c[10];

1. 掌握字符串的应用；

字符串实际上是使用空字符 **\0** 结尾的一维字符数组

|  |  |
| --- | --- |
| **序号** | **函数 & 目的** |
| 1 | **strcpy(s1, s2);** 复制字符串 s2 到字符串 s1。 |
| 2 | **strcat(s1, s2);** 连接字符串 s2 到字符串 s1 的末尾。 |
| 3 | **strlen(s1);** 返回字符串 s1 的长度。 |
| 4 | **strcmp(s1, s2);** 如果 s1 和 s2 是相同的，则返回 0；如果 s1<s2 则返回小于 0；如果 s1>s2 则返回大于 0。 |
| 5 | **strchr(s1, ch);** 返回一个指针，指向字符串 s1 中字符 ch 的第一次出现的位置。 |
| 6 | **strstr(s1, s2);** 返回一个指针，指向字符串 s1 中字符串 s2 的第一次出现的位置。 |

1. 熟悉数组元素的引用方法；

/\*调用数组元素的三种方法：下标法、数组名法、指针法\*/

#include<stdio.h>

int main()

{

int a[] = { 1,2,3,4,5 }, i, \* p;

printf("用下标引用数组元素：\n");

for (i = 0; i < 5; i++)

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

printf("用数组名引用数组元素：\n");

for (i = 0; i < 5; i++)

printf("%d\n", \* (a+i));

printf("用指针引用数组元素：\n");

for (p = a; p < a+5; p++)

printf("%d\n", \* p);

return 0;

}

1. 理解二维数组的定义及应用。

**类型名 数组名[ 行表达式 ][ 列表达式]；行与列用常量表达式。**

/\*\* 定义数组 \*/

int main()

{

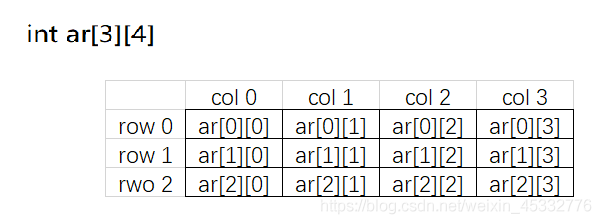
int ar[3][4]; // 3 行 4列 未初始化

char br[3][4];

double cr[3][4];

return 0;

}



# 6.模块化程序设计

（1）掌握函数的定义和调用方法；

1.函数的返回值类型

2.函数的名称

3.函数的参数

int wrongplus(int a, int b)

{

a = a + 1;

b = b + 1;

return a + b;

}

返回值类型 函数名字（形参列表）

{

//函数体，函数的功能在函数体里实现

}

变量=函数名（实参列表）；//带返回值的

函数名（实参列表）； //不带返回值的

（2）熟悉常用库函数的使用方法；

#include <math.h>数学函数

|  |  |
| --- | --- |
| int abs( int x) | 求整数x的绝对值 |
| double fabs(double x) | 求双精度实数x的绝对值 |
| double acos(double x) | 计算cos-1(x)的值 |
| double asin(double x) | 计算sin-1(x)的值 |

#include <ctype.h>字符函数

|  |  |
| --- | --- |
| nt isalnum(int ch) | 检查ch是否为字母或数字 |
| int isalpha(int ch) | 检查ch是否为字母 |
| int iscntrl(int ch) | 检查ch是否为控制字符 |

#include <string.h>字符串函数、

|  |  |
| --- | --- |
| char \*strcat(char \*s1,char \*s2) | 把字符串s2接到s1后面 |
| char \*strchr(char \*s,int ch) | 在s所指字符串中，找出第一次出现字符ch的位置 |
| int strcmp(char \*s1,char \*s2) | 对s1和s2所指字符串进行比较 |

#include <stdio.h>输入输出函数

|  |  |
| --- | --- |
| nt fclose(FILE \*fp) | 关闭fp所指的文件，释放文件缓冲区 |
| int feof (FILE \*fp) | 检查文件是否结束 |

#include <stdlib.h>动态分配函数和随机函数

|  |  |
| --- | --- |
| void \*realloc(void \*p,unsigned size) | 把p所指内存区的大小改为size个字节 |
| int rand(void) | 产生0～32767的随机整数 |

（3）熟悉形式参数与实际参数的概念；

在函数定义中出现的参数可以看做是一个占位符，它没有数据，只能等到函数被调用时接收传递进来的数据，所以称为**形式参数**，简称**形参**。

函数被调用时给出的参数包含了实实在在的数据，会被函数内部的代码使用，所以称为**实际参数**，简称**实参**。

4）熟悉局部变量和全局变量的概念；

局部变量也称临时变量，在函数、代码块内定义，一般只可在代码块内部使用的变量。

全局变量具有全局性，放在函数外，在同一\_\_\_.c文件中可在其定义之后的任何地方使用。

（5）了解函数的嵌套调用与递归调用。

嵌套调用是指在函数A()中调用函数B(),在函数B()中又调用函数C(),诸如此类的复杂调用.

#include <stdio.h>

//定义函数

int a1(int m) {

printf("%d的平方值为:%d", m, m \* m);

return 0;

}

int a2() {

int n;

printf("请输入一个数字:");

scanf\_s("%d", &n);

a1(n);

return 0;

}

int main(void) {

a2();

}

递归调用是指一个函数直接或者间接的调用自己本身.如果一个函数直接调用自己叫**直接递归**,如果一个函数简介调用自己叫**间接递归**.

#include <stdio.h>

//递归调用求a的n次方

double Pow(double a,int n) {

if (n == 0) return 1;

if (n == 1) return a;

return Pow(a, n / 2) \* Pow(a, n - n / 2);

}

int main(void) {

double a;

int n;

printf("求a的n次方值\n");

printf("输出a的值:");

scanf\_s("%lf", &a);

printf("输出n的值:");

scanf\_s("%d", &n);

printf("%.lf^%d = %.4f",a,n,Pow(a, n));

return 0;

}