C99和C11提供%zd转换说明匹配siezeof的返回类型。

一些不支持C99和C11的编译器可用%u或%lu代替%zd。

float f=(float)3.4 或float f = 3.4f

如果你在数据的后面加上一个f时，系统默认为float类型，此时就不需要系统转换类型，可以直接赋值给float类型定义的变量中。从而减少一些不必要的工作，提高代码的质量和运行速度。

%\*d %\*.\*f可以用来限定输入值的字段宽度

如scanf(“%\*d %\*.\*f \n”);

把\*放在%和转换字符之间时，会使得scanf（）跳过相应的输入项。在程序需要读取文件中特定列的内容时，这项跳过功能很有用。

a\_post=a++ //后缀：使用a的值后，递增a

b\_post=++b //前缀：使用b的值前，递增b

前缀形式：q=2\*++a 首先，a递增1；

然后，2\*a，并将结果赋给q

后缀形式：q=2\*a++ 首先，2\*a，并将结果赋给q

然后，a递增1

n++：“先使用n，再递增”

++n：“先递增n，再使用”

5.0/2 除法=2.5

5/2 取整=2

5%2 取余=1

% 判断奇偶时用此运算符

a=123；

x=a%10

y=a%100/10 或 a/10%10

z=a/100

利用符号取一个数的不同位的数值

Else与它前面最近的不带else的if相结合，与书写格式无关，若想改变用{}

Switch (表达式)

{

case 常量:

语句:break:

case 常量:

语句；break；

…………………

default:

语句;

}

（注：break; 从switch里跳出）

赋值语句为0

利用计算机求解问题的一般过程

1. 问题分析阶段
2. 数据结构设计阶段
3. 算法设计阶段
4. 编码与调试阶段

算法描述的必要性

程序设计过程：算法设计+程序实现

算法描述：

描述解题逻辑

独立于程序设计语言

程序实现：

利用程序设计语言的功能，实现算法

熟悉语言的语法、语义、支撑环境

算法描述方法

1. 流程图
2. 伪码

随机数：

#include<stdio.h>

#include<stdlib.h>

#include<time.h>

int main(void)

{

int a[10];

int i;

srand((unsigned)time(NULL));

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

a[i] = rand()%10;

}

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

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

}

return 0;

}

#include<stdio.h>

#include<string.h>

int main(void)

{

strcat(x,y); //将y连接至x末尾

strcmp(x,y); //比较x，y 相等为0，x大为1，y大为-1

strlen(x); //x字符串长度（'\0'占字符不占长度）

strcpy(x,y); //将y复制入x

}

a[i]等价于\*(a+i)

static 静态变量

stdlib.h

void \*malloc(int size) //申请

void free(void \*p) //释放

int\*ptr=(int\*)malloc(sizeof(int));

\*ptr=10;

Typedef struct date{

int year;

int month;

int day;

}DATE;

文件打开与关闭 fopen, folose

• 字符输入输出 fgetc, fputc

•格式化输入输出 fscanf, fprintf

•二进制输入输出 fread, fwrite

•字符串读写 fgets, fputs