## C语言概述

国际模糊C代码大赛 www.ioccc.org

C语言缺陷 可以参考 <<C陷阱与缺陷>>

C语言中缺少 bool类型变量 必须要要用 stdbool.h

以下划线后跟一个大写字母开头的名字是保留字 程序员不应该使用

练习题中的xxx页 指的是IPAD 横版的 页码

在C99 模式下才允许 在使用for循环时 这样定义:for(int I =0 ; i<10;i++);

在C中 在for中声明的i 不能在for循环体外使用 除非已经在for循环外已经声名.

字长:cpu一次能并行处理的二进制位数 一个二进制是8位 所以字长是8的倍数

因为指针涉及到寻址空间 所以与所处的位数有关 在32位机中是4字节 与指向的对象无关

Int \*p[] 指针数组 int (\*p)[] 指向数组的指针

## C语言基本概念

包括 预处理指令 函数 变量 语句

**预处理**:程序会被送交给预处理器.预处理器执行以#开头的命令.它可以给程序添加内容,也可以对程序进行修改.

**编译**:修改后的程序可以进入编译器,编译器会把程序翻译成机器指令 即 目标代码 然而这样的程序还是不可以运行

**链接**:链接器把由编译器产生的目标代码和所学的其他附加代码整合到一起,这样才能生成可执行的文件.

UNIX C编译器 为 cc

% cc pun.c

Gcc –o pun pun.c

### 2.1 指令

指令都是以字符#开始的 这个字符可以把C程序中的指令和其他代码区分开来,指令默认只占一行,每条指令的结尾没有分号或其他特殊标记

### 2.2 函数

**例程**:是某个系统对外提供的功能接口或服务的集合.

一个C程序可以包含多个函数 但只有main函数是必须有的.main函数是非常特殊的,在执行程序时系统会自动调用main函数.

### 2.3 语句

**语句**:程序运行时执行的命令.

由于语句可以连续占用多行,有时候很难确定它的结束位置,因此用分号来向编译器显示语句的结束位置.但指令通常只占一行,因此不需要分号结尾.

### 2.4 注释

一旦遇到 /\* 编译器读入 并且忽略 随后的内容直到遇到符号 \*/ 为止

### 2.5 变量和赋值

1.变量的类型

Float型变通常比int型变量慢 更重要的是 float型变量所存储的数值往往只是时机数值的一个近似值.

Float类型的值通常分成两部分存储 小数(尾数)部分和指数部分 12.0这个数可以以1.5X2^3存储 又称real类型

2.声明

在使用变量值钱必须对其进行声明(为编译器所做的描述) 指定变量的类型 说明变量名字

3.赋值

变量通过赋值的方式获得值,在同一个声明中可以对任意数量的变量进行初始化.

Int height=9,length=10

C语言的通用规则 在任何需要数值的地方都可以使用具有相同类型的表达式.

### 2.6 读入输入

scanf,printf

### 2.7 定义常量的名字

宏定义的特性给常量命名

#define INCHES\_PER\_POUND 166

宏定义表达式

#define RECIPROCAL\_OF\_PI (1.0f / 3.14159f)

### 2.8 标识符

标识符可以含有数字,字母和下划线 必须以字母或者下划线开头,C语言是区分大小写的.虽然说c对标识符的最大长度没有限制,不用担心使用较长的描述性名字 但是C语言却之区分前31个字符 如果两个标识符前31个字符是一样的那么可能区分不开

关键字 请参见第39页.

Token 记号 在不改变意思的基础上无法再分割的字符组

GNU 是自由软件基金会的一个项目 自由软件基金会 旨在抗议对UNIX软件授权的各种限制.

**练习题参见 43页**

## 格式化输入/输出

Scanf和printf 用来格式化输入和输出.

### 3.1 printf函数

Printf(格式串,表达式1,表达式2); 格式串包含普通字符和转换说明 转换说明以%开头.c语言中不会检测格式串中转换说明的数量是否和输出项的数量相匹配.

转换说明可以用%m.pX格式或%-m.pX格式.

M:制定了显示的最少字符数量.如果要显示的数值所学的字符数少于m 那么值在字段内是右对齐的,若显示的数值多于m,那么字段会自动扩展为所需的尺寸,在m前加上一个负号会导致左对齐.

X:表明在显示数值前需要对齐进行哪种转换

在printf 显示% 连续输入%%那么它将显示一个字符%

### 3.2 scanf 函数

scanf(‘%d’,&i); 调用scanf函数是读数据的一种有效但不理想的方法.

**练习题参见54页**

## 表达式

条件运算符的优先级低于先前介绍过的所有运算符 只比赋值运算符高

左值:表示存储在计算机内存中的对象而不是常量或计算的结果.

在表达式后添加分号 将其转换为语句.

### 4.1 算数运算符

一元算术运算符: -(负号) +(正号) 都是右结合的

二元算术运算符: +(加法)-(减法) \*(乘法) /(除法) %(求余) 都是左结合的

### 4.2 关系运算符

> < <= >= 优先级低于 算数运算符 左结合

== != 判等运算符 低于 关系运算符 左结合

### 4.3 逻辑运算符(短路计算)

一元:! 优先级和一元正负号相同 右结合

二元:&& || 优先级 低于关系运算符和判等运算符 左结合

### 4.4 赋值运算符

简单赋值: = 右结合

复合赋值: += -= \*= 右结合

左操作数必须是左值

### 4.5 自增自减运算符

后缀的++ 后缀的— 比一元的正号 负号优先级高 左结合

前缀的++ 前缀的— 和一元的优先级相同 都是右结合

### 4.5 运算符的优先级和结合性

左结合:从左向右结合的.

右结合:从右向左结合

**练习题参见:68页**

## 语句

### 5.1 选择语句

If else , Switch

Switch语句往往比if语句执行的速度快

Case 常量表达式 不能包含变量和函数调用

为了节省空间 case4: case3: case 2: 可以放在同一行

### 5.2 重复语句

While , do , for

**练习题见105页面**

### 5.3 跳转语句

Break

Continue

Goto

### 5.4 空语句

结尾除了分号以外什么都没有的 语句

### 5.5 复合语句

{} 在周围防止花括号 可以强制变异将其作为一条语句来处理

**练习题 85页**

## 基本类型

### 6.1 整数类型

分为有符号和无符号 默认为有符号

有符号:-2^15~2^15-1 无符号:2^16-1

八进制和十六进制只是书写数的方式 它们不会对数的实际存储产生影响 都是以二进制形式存储的

0xffffffffful 无符号长整型 字母l,u的顺序和大小写无关

读写短整数时 在d 前面加上字母h

读写长整数时 在 d 前面加上字母l

### 6.2 浮点类型

Float 单精度浮点数, double 双精度浮点数 . long double 扩展精度浮点数

C99 标准中 浮点类型扥为两种 一种是 实浮点型 另一种是复数类型 包括 float\_complex

### 6.3 字符类型

ascII码

字符常量需要用单引号括起来 而不是双引号

c语言中把字符当作小整数进行处理 所有字符都是以二进制形式进行编码 字符常量是int类型 而不是char类型

char 类型 也存在有符号 和 无符号 两种 有符号 范围是 -128-127 无符号为0-255

字符处理函数:

toupper函数 再被调用时检测参数把参数转换为相应的大写字母

可以用getchar和putchar 替代 scanf和printf

getchar 返回的是一个int类型的值和scanf 函数一样 getchar函数也不会在读取时跳过空白字符 执行程序时getchar 函数胜于 scanf可以节约时间

### 6.4 转义序列(非打印字符)

分为字符转义序列和数字转义序列

### 6.5 类型转换

显式转换和隐式转换

Typedef int Bool sizeof 程序允许存储指定类型的所需的空间大小

类型定义和宏定义是两个重要的不同点.类型定义比宏定义更强大,数组和指针是不能定义为宏的

#defined PTR\_TO\_INT int \*;

PTR\_TO\_INT p,q,r 会变成 int \* p,q,r 这里只有p是指针 但是类型定义不会有这样的问题

Typedef 命名的对象 具有和变量相同的作用域规则 定义在函数体内的typedef名字在函数外是无法识别的.另一方面 宏定义的名字在预处理时会在任何出现的地方替换掉

**练习题见129**

## 数组

聚合类型:数组和结构

数组:含有多个数据值的数据结构,并且每个数据值具有相同的数据类型.

### 7.1 一维数组

声明数组 需要指明数组元素的类型和数量 int a[10 ] 比较好的方法是用 比较好的方法是用宏来定义数组的长度

数组初始化: int a[10] = {1,2,3,4,5,6,7,8,9}

如果初始化式比数组段 那么数组中剩余的元素 为0

如果初始化式为空是非法的 所以在大括号内放上一个0 初始化式比要初始化的数组长也是非法的

如果定义了初始化式 可以省略数组的长度

C99指定初始化式

Int a[10] = {[2]=22,[4]=44};

Sizeof 计算字节数字 通过 sizeof 计算数组的长度 sizeof(a) /sizeof(a[0])

### 7.2 多维数组

虽然在平常画多维数组 是使用数学矩阵 但是在内存中是按照行主序存储数组的

多维数组初始化 int [2][2] = {{1,1},

{2,2}}

C语言数组 为什么从0下标 开始 因为 地址加0 正好是数组地址 不用计算1-1的偏移量

**练习题见144**

## 函数

每一个形式参数 都必须有类型

函数定义 返回类型 函数名 (参数类型 形参名字){

声明语句;

}

当函数的调用 在函数的声明之前 编辑器不会报错 而是隐式的声明为init类型 函数

在调用前声明每个函数

实际参数 是通过值传递的

函数不能返回数组

编译器把不跟圆括号的函数名看成是指向函数的指针.指向函数的指针有合法的应用

**练习题见 170页**

## 程序结构

### 9.1 局部变量

在函数体内声明的变量为局部变量,该变量的存储单元是在函数调用时自动分配的 函数返回时收回分配

静态局部变量: static 静态存储期限 静态存储期限的变量拥有永久的存储单元变量

静态局部变量和非静态局部变量都是块作用域的 静态变量 在函数返回时不会释放 下次再次调用函数时 还会存储该值

### 9.2 外部变量(全局变量)

静态存储期限

文件作用域:从变量被声明的点开始一直到所在文件的末尾 因此 跟随在外部变量声明之后的所有函数都可以访问

### 9.3 程序块

默认情况下 生命在程序块中的变量的存储期限是自动的.

### 9.4 作用域

#include<stdio.h>

Int i=100;

Void f(int i){

i=1;

}

Void g(void){

int i=2;

if(i>0){

Int I;

I =3;

}

i=4;

}

Void h(void){

i =5;

}

**练习题 189页**

## 指针(非常重要)

10.1 指针变量

现代计算机都将内存分割为字节 存储8位

指针就是地址 指针变量 就是存储地址的变量

在指针变量名字前 放置\* 如 int \*p

取地址符号 & 那么 &x 就是x在内存中的地址 间接寻址 \*p

Printf 函数 格式串 使用转换说明%p 来表示

用变量p存储变量i的地址 p指向i

**练习题201 页**

## 第十一章 指针和数组 (需要多看几遍)

当指针指向数组元素时,C语言允许对指针进行算术运算

支持三种格式的指针算术运算 指针加上整数 指针减去整数 两个指针相减

指针的比较 只有在两个指针指向同一数组时 用关系运算符进行的指针比较才有意义

指向复合常量的指针 int \*p = (int[]){3,0,3,4,1};

\*p++= j 因为后缀++的优先级高于\* 所以上述 \*(P++)=j

int a[10];

\*a =7 把7存储在a[0]

\*(a+1) = 12 把12存储在a[1] a+I 等同于&a[i]

#define N 100

Int a[N],I,sum = 0,\*p=a;

For(i=0;i<N;I++)

sum+=p[i]; p[i] 看做是\*(p+i)

**练习题215页**

1. **14,34,4,真,假**

## 第十二章 字符串

字符串字面量

当C语言编译器在程序中遇到长度为n的字符串字面量时,它会为字符串字面量分配长度n+1的内存空间.这块内存空间将用来存储字符串字面量中的字符,以及一个用来标识字符串末尾的额外字符 \0

空字符的码值为0 零字符则有不同的码值

char date1[8] = “June 14”;

编译器将把字符串”june 14”中的字符复制到数组date1中,然后追加一个空字符从而使date1可以作为字符串使用.c编译器会把它看成是数组初始化式的缩写形式 char date1[8] = {j,u,n,e,’’,1,4,\0}

如果初始化式太短以至于不能填满字符串,编辑器会添加空字符.

在声明为数组时,就像任意数组元素一样 可以修改存储在date中的字符,在声明为指针时,date指向字符串字面量,字符串字面量是不可以修改的,声明数组时 date是数组名 在声明为指针时 date是变量