

821

TP311.1
L71

中国计算机函授学院图书编写中心 组编

全国高等教育自学考试应试指导丛书

计算机及应用专业(专科)

高级语言程序设计自考应试指导

主 编 刘 锋

副主编 陈 兵

主 审 牛允鹏

南 京 大 学 出 版 社

内 容 简 介

本书严格按照“全国高等教育自学考试指导委员会”制定的“计算机及应用专业(专科)”《高级语言程序设计自学考试大纲》和组编的教材《高级语言程序设计》编写

本书分为四个部分。第一部分介绍了教材各章节的知识点、典型题的分析与解答、章末习题解答与分析;第二部分给出了详细的实验指导;第三部分给出了两套模拟试卷及分析解答;第四部分是附录,给出了C语言的语法汇总、常用系统函数汇总、编译错误汇总。

本书是参加《高级语言程序设计》课程自学考试考生的必备辅导材料,也可以作为辅导老师和在校学生的教学参考书。

图书在版编目(CIP)数据

高级语言程序设计/刘锋主编. —南京:南京大学出版社, 2001.2

INBS 7 - 305 - 02159 - 8

I . 高... II . 刘... III . 程序语言-程序设计 IV . TP312

中国版本图书馆 CIP 数据核字(2001)第 69015 号

书 名 高级语言程序设计自考应试指导

主 编 刘 锋

副 主 编 陈 兵

主 审 牛允鹏

丛书主编 牛允鹏 胡学联

责任编辑 蒋劲柏

出版发行 南京大学出版社

地 址 南京汉口路 22 号 邮编 210093 电话 025 - 3593695

印 刷 合肥学苑印刷厂

经 销 全国各地新华书店

开 本 787 × 1092 1/16 印张 17 字数 425 千

版 次 2001 年 3 月第 1 版 2001 年 4 月第 2 次印刷

定 价 24.00 元

ISBN 7 - 305 - 02159 - 8/TP · 208

声明:(1)版权所有,侵权必究。

(2)本版书若有质量问题,可向经销商调换。

第一部分

知识点概述、典型题解 与章末习题解答

本部分以教材中的章为单位,依次介绍了自考大纲中各个知识点、典型题的分析与解答、教材中章末习题分析与答案。

在“知识点”部分主要介绍了各个知识点的内容,对重点和难点进行了简要的分析。

在“典型题的分析与解答”中精选了部分有代表性的题目进行了解答过程的分析,最后给出正确的参考答案。

在“教材章末习题分析与解答”中给出了所有章末习题的解答分析过程,最后给出正确的参考答案。

第 1 章 概 述

1.1 知 识 点

1. 程序、程序设计和高级语言(达到“识记”层次)

(1) 程序的概念

程序是用语言、文字、图表等方式表达解决某个问题的方法步骤(算法)。计算机程序是用指定的计算机语言表达解决某个问题的方法步骤(算法),这些方法步骤必须是计算机能理解并且能执行的。

(2) 程序设计的任务和主要步骤

程序设计主要指研制计算机程序的过程,它应该包含:对问题的分析与了解、选取解决该问题的方法步骤(算法)、用特定的计算机语言将方法步骤记录下来,记录下来的这个方法步骤就是程序清单。

(3) 高级语言的概念与作用

计算机语言有很多种,按照大类来分,有面向机器的语言,也称低级语言(比如二进制语言,也称机器语言;符号语言,也称汇编语言);有面向问题的语言,也称高级语言(比如 C 语言、PASCAL 语言、ALGOL 语言、FORTRAN 语言、BASIC 语言等)。

低级语言中的机器语言就是二进制语言,用这种语言编写的程序可以在特定的计算机上立即执行,所以称为可执行的目标程序。其他的低级语言(符号语言、汇编语言)必须通过名为“汇编程序”的软件将其翻译成二进制语言的目标程序,才能执行。

用高级语言编写的程序一般称为“源程序”,必须翻译成目标程序才能执行。完成这个翻译任务的软件有两种:一种是翻译一句,执行一句,完成翻译任务的软件称为“解释程序”,比如一般的 BASIC 语言编写的源程序就是通过“BASIC 解释程序”翻译并执行的,这种源程序是没有目标程序的;另一种是一次翻译成目标程序,完成翻译任务的软件称为“编译程序”,例如 C 语言编写的源程序就要通过名为“C 语言编译程序”将其翻译成目标程序,然后再执行目标程序,这种源程序是有目标程序的。

2. C 语言和 C 程序(达到“识记”层次)

(1) C 语言的发展

C 语言的前身是 ALGOL 语言。63 年英国剑桥大学在 ALGOL 语言基础上增添了硬件处理能力后,推出了“CPL”(复合程序设计语言)语言;67 年剑桥大学的马丁·理查德对其简

化,提出“BCPL”语言;70年美国贝尔实验室的肯·汤普逊进一步简化,提出“B”语言(BCPL的第一个字母);72年贝尔实验室的布郎·W·卡尼汉和丹尼斯·M·利奇对其完善和扩充,提出了“C”语言(BCPL的第二个字母);87年美国标准化协会制定了C语言标准,称为“ANSI C”语言,也就是现在流行的C语言。

(2)C 程序的基本结构

用C语言编写的程序是有其规定的结构形式的,一般来说,C语言的源程序是由若干个函数组成的,其中有且仅有一个“主函数”,其余函数称为“非主函数”。每个函数包括“函数头”和“函数体”两部分。所有非主函数可以相互调用,但是不能调用主函数;主函数可以调用任何非主函数,但是不能调用自己(主函数)。程序中可以出现注释,需用“/*”和“*/”将注释包括起来。“/*”、“*/”及其间的注释只是起到对程序说明的作用,以便于阅读和理解程序、程序段、语句的作用,编译时,所有的注释将被跳过。

3.C 语言的字符集、词类和句类(达到“领会”层次)

(1)C 语言的字符集

C语言的字符集是用来书写源程序清单时允许出现的所有字符的集合,也就是说,在C语言的源程序清单中不能出现字符集以外的字符。注意,字符集主要是由键盘符号,即普通键盘上白色键组成的,对于普通键盘上的灰色键,由于其代表的是一种操作控制,没有对应的符号,在程序中需要通过特殊方式来表示这些键,这种特殊方式就是“转义字符”。此外,如果使用了汉字操作系统,汉字就相当于英文字母,也可以出现在源程序清单中。

(2)C 语言的词类

C语言的字符是用来构成词汇的。C语言的单词主要分为6大类:常量、变量、运算符、函数调用、表达式、保留字。

常量可以有名称,变量、函数必须有名称,这些名称的命名必须遵守一定的规则,这个规则就是“标识符”,它是由字母或者下划线开头的字母、数字、下划线组成的一串字符,其中字符的数目不得超过32个,在微型机上只有前8个有效。

(3)C 语言的保留字

为了在程序中表达某个特定的语法含义,系统保留了32个特别的标识符,这些特别的标识符称为“保留字”,也称为“关键字”。用户在给常量、变量、函数等对象起名称时,不得使用这些保留字。

(4)C 语言的语句

词类是用来构造语句的。C语言的语句主要分为8大类:数据定义语句、赋值语句、函数调用语句、表达式语句、流程控制语句、复合语句、空语句、其他语句。对每种语句来说,有它规定的书写格式,称为“语法结构”,有它规定完成的任务,称为“语句功能”。

语句是用来构造程序的,用户在程序中选择语句时,首先要考虑语句的功能,其次要遵守其语法结构。忽视前者,程序将无法完成既定的任务;忽视后者,程序通不过语法检查,则不能翻译成可执行的目标程序。

1.2 典型例题分析与解答

1. 用高级语言编写的源程序必须通过_____程序翻译成二进制程序才能执行, 这个二进制程序称为_____程序。

【分析】用高级语言编写的源程序有两种执行方式: 一是利用“解释程序”, 翻译一条语句, 执行一条语句, 这种方式不会产生可以执行的二进制程序, 例如 BASIC 语言; 二是利用“编译程序”一次翻译形成可以执行的二进制程序, 例如 C 语言。凡是编译后形成的可执行二进制程序都叫做“目标程序”。

【答案】编译
目标

2. 关于 C 语言, 下列说法中错误的是()

- ①数据类型丰富, 程序设计简单
- ②运算符丰富, 算法描述方便
- ③是一种结构化的程序设计语言
- ④语法结构复杂, 难学

【分析】C 语言只有 8 类语句, 语法简单易学。

【答案】④

3. 下列符号中, 不属于转义字符的是()

- ① \\\
- ② \x00
- ③ \00
- ④ \09

【分析】只要是以“\”开头的、后跟指定单个字符的都是“转义字符”。具体能跟哪些符号要参看教材 P4 的表 1-2。还要注意“转义字符”的另一种方式, 在“\”后面跟一个 8 进制数或 16 进制数。如果是 16 进制数, 前面要加字母“x”。按照上述规则, 第①个可选答案中应为 8 进制数, 但是其中出现了非法数字“9”, 所以该答案不是转义字符。

【答案】④

4. 下列符号串中, 不属于 C 语言保留字的是()

- ① if
- ② then
- ③ static
- ④ for

【分析】参看教材中 P5 页, 可以知道“then”不是 C 语言规定的保留字。

【答案】②

5. 下列符号串中, 属于 C 语言合法标识符的是()

- ① else
- ② a-2
- ③ _00
- ④ 00_

【分析】标识符必须是“字母”或者“下划线”开头的, 由字母、数字、下划线组成的一串符号, 并且不能是保留字。

【答案】③

6. 形如_____的式子称为“函数调用”。

【分析】函数调用的式子规定为“函数名(实际参数表)”。如果作为“词类”来调用,则该式子代表执行函数后返回的一个值;如果作为“语句”来调用,则代表执行一次函数。

【答案】函数名(实际参数表)

7. 表示简单加工计算的符号称为_____。

【分析】运算符是代表一种简单的加工运算的符号。

【答案】运算符

8. 下列说法中,正确的是()

- ①主函数名是由程序设计人员按照“标识符”的规则选取的
- ②分号和回车符号都可以作为一个语句的结束符号
- ③在程序清单的任何地方都可以插入一个或多个空格符号
- ④程序的执行,总是从源程序清单的第 1 行开始

【分析】主函数名规定为“main”,不是程序设计人员自由选取的;分号是表示语句结束的唯一符号;程序的执行总是由主函数的第 1 条语句开始执行的。

【答案】③

1.3 教材章末习题分析与解答

一、单项选择题

1.

【分析】二进制语言是低级语言,低级语言又称为“面向机器的语言”。

【答案】①

2.

【分析】高级语言又称为“面向问题的语言”,C 语言和 ALGOL 语言都是高级语言,也是面向问题的语言。

【答案】②

3.

【分析】一般来说,以“\”开头的都属于“转义字符”,而可选答案①中不是转义字符,仅仅是单个“\”,显然是键盘符号。

【答案】①

4.

【分析】转义字符必须是以“\”开头的单个字符或更多个字符(8 进制数或 16 进制数)。

【答案】③

5.

【分析】C 语言的保留字也称“关键字”,“char”属于 C 语言的保留字,但是“cher”不是保留字。

【答案】②

6.

【分析】标识符的构成规则是:以字母或者下划线开头的,由字母、数字、下划线组成的,长度不超过 32 个字符(在 PC 机中为前 8 个字符有效)的符号串。

【答案】④

7.

【分析】任何 C 语言的程序都是由主函数的第 1 条语句开始执行,并且在主函数中结束程序的运行。如果程序中出现了非主函数,则一定是通过主函数或其他非主函数的调用才能执行。

【答案】④

二、填充题

1.

【分析】二进制语言、符号语言、汇编语言都是低级语言,低级语言称为面向机器的语言;高级语言称为面向问题的语言。

【答案】机器 问题

2.

【分析】用高级语言编写的程序称为源程序,有两种方式来执行高级语言编写的源程序:一是利用“解释程序”,翻译一条语句,执行一条语句;二是利用“编译程序”,一次将源程序中的所有语句全部翻译成二进制程序(称为目标程序),然后再执行这个目标程序。

【答案】源 解释 编译 目标

3.

【分析】转义字符总是由“\”开始,后跟指定的某些单个字符来代表 C 语言字符集中的某些特定符号(主要是键盘上灰色键对应的控制符号);也可以跟一个 8 进制数,或者以“x”开始的一个 16 进制数。

【答案】反斜杠(\)

4.

【分析】C 语言的词类主要有 6 大类,具体请参看教材。其中的常量、变量、运算符、函数调用可以用来构成表达式,保留字是具有特定语法含义的符号串。

【答案】常量 变量 运算符 函数调用 表达式 保留字

5.

【分析】C 语言的语句分为 8 大类,具体请参看教材。其中的函数调用、表达式既可以构成词类,也可以构成语句,关键在于其后是否有“分号”,有则为语句,无则为词类。

【答案】数据定义 赋值 函数调用 表达式 流程控制 复合 空 其他

6.

【分析】C 程序中必须有主函数,而且只能有 1 个主函数。执行时,总是从主函数开始,并在主函数中结束。程序中的非主函数要么是主函数调用、要么是其他非主函数调用才能执行到。

【答案】1 主 主

7.

【分析】C 语言规定:注释是从“/*”开始,到“*/”结束的中间符号串。

【答案】/* */

第 2 章 基本数据类型和运算

2.1 知 识 点

1. 数据类型的概念及其分类(达到“识记”层次)

(1)数据类型的概念

每个数据都有特定的加工方式,比如整数或实数是可以进行各种算术运算和加工的,而姓名是可以按照字典顺序比较大小的,数据的这种特定加工方式就称为数据的类型,简称“数据类型”。

(2)数据类型的分类

C 语言规定数据类型分为:基本数据类型、构造类型、指针类型、枚举类型、空类型。

其中最基本也是最常用的是基本数据类型,它包括整型、实型和字符型。整型数据中又分为带符号的和不带符号的一般整型、短整型、长整型;实型中又分为单精度和双精度实型。

构造类型是由其他数据类型组合而成的数据类型,分为数组、结构型和共用型三种。

指针型数据是 C 语言特有的一种数据类型,这种类型的数据就是内存单元的地址。

枚举型数据是用便于阅读的标识符来代表特定的几个数据,以利于阅读程序。

空类型是指没有数据类型的数据,只在特定场合中使用。

对于读者来说,要特别注意每种类型的数据在内存中占用的字节数目,这个字节数称为“数据长度”。

2. 基本类型常量的书写方法(达到“领会”层次)

常量是程序运行中不改变的数据,为了简单起见,常量通常就在程序中直接写出来。

(1)整型常量的书写方法

整型常量的书写方法很简单,只要写成整数就可以。如果前面有“-”或“+”,则肯定是带符号数,如果前面没有“-”和“+”,则既可以看成不带符号数,也可以看成带符号数(正数)。书写整型常量时,还要注意,如果整数的后面有符号“L”或“l”,则表示是长整型数据,否则是短整型数据。

(2)实型常量的书写方法

实型常量有两种书写方法:一种是通常的实数写法,一种是指数写法。注意,书写实型常量时,是不区分“单精度”和“双精度”的,一律按照单精度数据来理解,其在内存中占用 4 个字节。

(3) 字符型常量的书写方法

字符型常量就是用单引号“'”括住的单个字符,包括转义字符。

(4) 字符串常量的书写方法

字符串常量就是用双引号“”括住的若干个字符,其中可以出现转义字符。

(5) 符号常量的定义方法

符号常量的数据类型可以是整型、实型、字符型。符号常量需要通过下列命令事先定义:

```
# define 符号常量 常量
```

其中的符号常量是一个由用户自己选取的标识符,一般情况下建议用大写字母构成,其值就是定义中的“常量”,符号常量的数据类型就是“常量”的数据类型。

3. 基本类型变量的定义、初始化方法(达到“领会”层次)

所有基本类型变量的定义和初始化都是通过下列的“数据定义语句”来进行的:

```
存储类型 数据类型 变量名 = 初值, ...;
```

其中的“存储类型”可以选取下列 4 种存储类型符:auto(自动型)、register(寄存器型)、static(静态型)、extern(外部型)。选取那种“存储类型”可以参看下面第 5 点中的介绍。

其中的“数据类型”可以选取下列 9 种数据类型符:int(整型)、short(短整)、long(长整)、unsigned int(无符号整型)、unsigned short(无符号短整)、unsigned long(无符号长整)、float(单精度实型)、double(双精度实型)、char(字符型)。

其中的“变量名”是用户自由选取的标识符。

其中的“初值”是常量或者常量和运算符组成的常量表达式。

4. 基本类型数据在内存中的存放方式(达到“识记”层次)

数据(包括常量和变量)都需要在内存中存放,也就是说,每个数据都要分配一定数量的内存单元给它。相同数据类型的数据(不管是常量还是变量),在内存中的存放方式是完全相同的;不同数据类型的数据,在内存单元中存放的方式是不同的。作为一个程序设计人员,需要了解各种数据类型的数据在内存中的存放方式。

(1) 整型数据的存放

整型数据在内存中存放的是二进制整数,一般说,总是高位在后,低位在前的顺序。整型、短整型数据存放的是 2 字节的整数、长整型数据存放的是 4 字节的整数。如果是带符号整型数据,则正数存放的是原码,负数存放的是补码。

(2) 实型数据的存放

实型数据在内存中存放的是二进制的浮点数,一般的格式是“尾数 阶码”,其中的“尾数”和“阶码”都是一个带符号的整数形式。单精度实型数据占用 4 个字节,双精度实型数据占用 8 个字节。

(3) 字符型数据的存放

字符型数据在内存中存放的是该字符对应的 ASCII 代码值(8 位二进制无符号整数),每个字符型数据只占用 1 个字节。

(4) 字符串型数据的存放

字符串型数据在内存中存放的是每个字符对应的 ASCII 代码值(8 位二进制无符号整数),所占用的字节数是所有字符数目加 1。多出的这个字节存放的是字符串结束标记符号‘\0’,该字符的 ASCII 代码值为 00000000,即十进制的 0。

5. 变量的存储类型及其定义方法(达到“领会”层次)

所有变量的存储类型是通过“数据定义语句”中的“存储类型符”确定的。

变量的存储类型主要是指变量所分配的内存区在何处,此外,还可以用来说明在哪儿可以找到这个变量的定义。

解决变量存放在何处的“存储类型符”有下列三种:

auto	自动型	数据分配在内存堆栈区,这部分内存可以重复使用。
register	寄存器型	数据分配在 CPU(主机)中的寄存器中,可以重复使用。
static	静态型	数据分配在内存的数据区,这部分内存不能重复使用。

解决变量是在哪儿定义的“存储类型符”只有一个:

extern	外部型	用于多个编译单位之间数据的传递。例如在数据说明中,表示本编译单位中的未定义的变量是其他编译单位中定义的。
--------	-----	--

提个醒

- 真正掌握变量存储类型的使用,需要在函数一章中解决。

6. 变量的生存期和作用域(达到“领会”层次)

关于变量的分类有很多种。

从存储类型来说,分为自动型变量、寄存器型变量、静态型变量、外部参照型变量。

从数据类型来说,分为整型变量、实型变量、字符型变量等等。

从定义变量的位置来说,分为“内部变量”和“外部变量”。在函数体中定义的变量一律称为“内部变量”,在两个函数之间定义的变量一律称为“外部变量”。

从程序运行过程中变量是否活着(生存期)来说,分为“全局变量”和“局部变量”。从程序开始运行直到程序运行结束期间,变量都活着的变量称为“全局变量”,否则,称为局部变量。全局变量只有两种:外部变量和内部静态型变量。局部变量也有两种:内部自动型变量和内部寄存器型变量。

我们把变量能够使用的区域称为变量的作用域。变量的作用域主要取决于是内部变量还是外部变量。内部变量的作用域就是定义变量的函数或复合语句的内部;外部变量的作用域是从定义点到程序的结尾。

提个醒

- 不能错误理解全局变量的作用域是全局的,局部变量的作用域是局部的。
- 变量的作用域只与变量定义的地点有关,在所有函数外定义的变量(外部变量)的作用域是定义点到源程序的结尾,在函数或复合语句中定义的变量(内部变量)的作用域就是该函数或复合语句内部。
- 局部静态变量的生存期是全局的,作用域是局部的。

7. 基本运算符的运算规则和优先级(达到“简单应用”层次)

(1) 算术运算符

算术运算符分为常规算术运算符和增 1 减 1 运算符两类。

常规算术运算符有取正(+、单目前缀)、取负(-、单目前缀);加(+、双目中缀)、减(-、双目中缀)、乘(*、双目中缀)、除(/、双目中缀)、模(%、双目中缀)。算术运算符的运算对象为整型、实型,还包括字符型数据。运算规则就是普通的算术运算规则,需要注意的有两点:对除法运算来说,当两个对象都是整型或字符型时,结果为整型;对模运算来说,运算的结果是整除后的余数,结果的符号和被除数符号一致。

增 1 减 1 运算符有增 1(++、单目前缀或单目后缀)、减 1(--、单目前缀或单目后缀)。运算对象为整型、实型或字符型变量等。运算规则为对象值加 1 或减 1,当为前缀运算符时,对象先增(减)1 后,再参加运算;当为后缀运算符时,对象先参加运算后,再增(减)1。

提个醒

- 增 1 减 1 运算符只能施加于变量。

(2) 关系和逻辑运算符

关系运算符有小于(<、双目中缀)、小于或等于(<=、双目中缀)、大于(>、双目中缀)、大于或等于(>=、双目中缀)、等于(==、双目中缀)、不等于(!=、双目中缀)。运算对象是整型、实型或字符型数据。运算规则是逻辑判断,如果该式子成立,则运算结果值为整型的 1(表示真);否则为整型的 0(表示假)。

逻辑运算符有逻辑非(!、单目前缀)、逻辑与(&&、双目中缀)、逻辑或(||、双目中缀)。运算对象是整型(包括结果为整型的关系运算结果)、实型、字符型数据。运算规则请参看教材 P23 的表 2-8。

提个醒

- 关系和逻辑运算的结果都是整型:关系成立或逻辑真,结果为整数 1;否则为整数 0。
- 对于参与逻辑运算的对象,一律看作数值,非 0 看成逻辑真,0 看成逻辑假。

(3) 赋值运算符

赋值运算符分为一般赋值运算符和自反赋值运算符两类。

一般赋值运算符只有 1 个,就是赋值运算符(=、双目中缀)。前一个运算对象必须是变量,后一个运算对象是表达式。运算规则是先计算右边表达式的值,然后赋予左边的变量。

自反赋值运算符分为算术自反赋值运算符和位自反赋值运算符。算术自反赋值运算符有加赋值(+=、双目中缀)、减赋值(-=、双目中缀)、乘赋值(*=、双目中缀)、除赋值(/=、双目中缀)、模赋值(%=、双目中缀)。位自反赋值运算符参看下面的“位运算符”部分。前一个运算对象必须是变量,后一个运算对象是表达式。运算规则是先计算右边表达式的值,再和左边变量进行相应的运算,然后将结果赋予左边的变量。

(4)逗号、条件和长度运算符

逗号运算符“,”是双目中缀运算符,运算对象是整型、实型、字符型数据(通常都是表达式)。运算规则是先计算前一个对象(表达式)的值,然后计算后一个对象(表达式)的值。

条件运算符是“?”和“:”组成的,该运算符有三个运算对象,在“?”前面的运算对象是表达式 1,当做逻辑表达式看待;在“?”和“:”之间的运算对象是表达式 2,在“:”后面的运算对象是表达式 3。运算规则是判断“表达式 1”的值,若为非 0(真),则运算结果是“表达式 2”,否则运算结果是“表达式 3”。

长度运算符是“sizeof()”,运算对象必须放在圆括号中,运算对象可以是常量、变量、表达式,也可以是数据类型符。运算结果是测试对象占用的字节数,也就是对象的数据长度。

(5)位运算符

位运算符分为常规位运算符和位自反赋值运算符两类。

常规位运算符有位非(~、单目前缀)、位与(&、双目中缀)、位或(|、双目中缀)、位按位加(^、双目中缀)、位左移(<<、双目中缀)、位右移(>>、双目中缀)。运算对象一般是整型或字符型数据,常用的是无符号整型数据。运算规则可以参看教材 P28 的表 2-14、表 2-15 和表 2-16。

位自反赋值运算符有位与赋值(&=、双目中缀)、位或赋值(|=、双目中缀)、位按位加赋值(^=、双目中缀)、位左移赋值(<<=、双目中缀)、位右移赋值(>>=、双目中缀)。前一个运算对象必须是变量,后一个运算对象是表达式。运算规则是先计算右边表达式的值,再和左边变量进行相应的位运算,然后将结果赋予左边的变量。

提个醒

- 对于每个运算符,要注意能正确写出运算符号、正确选择运算对象、正确理解并运用运算规则。

- 对于不同的运算符,要注意运算符的优先级别,请读者参看教材 P207 的附录二。

- 对于同一级别的运算符,要注意运算符的结合性,请读者参看教材 P207 的附录二。

8. 表达式的构成规则及其优先级别(达到“简单应用”层次)

表达式的一般构成规则是用运算符将运算对象连接成的、正确的运算式子。要注意任何运算符连接正确的运算对象形成的运算式都是表达式。

表达式的一般运算规则是按照运算符的优先级和结合性依次对运算对象进行运算。

表达式值是按照运算规则进行运算后获得的一个最终结果值,注意任何表达式都有值。

表达式类型就是表达式值的类型。要注意由于在计算机中都是用二进制数来表示数据的,所以任何类型的表达式最终值都是数值。例如结果为整型、实型、字符型、逻辑值(1 或 0)等都是数值。由此看出,当任何表达式值为非 0 时,都可以看成是逻辑上的真,任何表达式值为 0,都可以看成逻辑上的假。

(1)算术表达式

算术表达式的构成是由算术运算符连接运算对象组成的运算式。算术表达式的运算必须按照运算符的优先级别和结合性依次计算,最终运算结果是一个算术值,即整型或实型数据。

(2)关系和逻辑表达式

关系表达式的构成是由关系运算符连接运算对象组成的运算式。关系表达式的运算必须按照运算符的优先级别和结合性依次计算,最终运算结果是一个整型值,即关系式成立,结果为 1,关系式不成立,结果为 0。

逻辑表达式的构成是由逻辑运算符连接运算对象组成的运算式。逻辑表达式的运算必须按照运算符的优先级别和结合性依次计算,最终运算结果是一个整型值,即 1 表示逻辑结果为真,0 表示逻辑结果为假。

(3)赋值表达式

赋值表达式的构成是由赋值运算符连接运算对象组成的运算式。赋值表达式的运算必须按照运算符的优先级别和结合性依次计算,最终运算结果等于赋予赋值运算符左边变量的值,可能是一个整型、实型、字符型数据,也可能是一个关系运算结果或逻辑运算结果(整型数据)。

(4)逗号表达式和条件表达式

逗号表达式的构成是由逗号运算符连接运算对象组成的运算式。逗号表达式的运算必须按照运算符的优先级别和结合性依次计算,最终运算结果是后一个表达式的值,可能是一个整型、实型或字符型数据,也可能是一个关系运算结果或逻辑运算结果(整型数据)。

条件表达式的构成是由条件运算符连接运算对象组成的运算式。条件表达式的运算必须按照运算符的优先级别和结合性依次计算,最终运算结果取决于第 1 个表达式的值:若第 1 个表达式值为非 0(真),则结果为第 2 个表达式值;若第 1 个表达式值为 0(假),则结果为第 3 个表达式值。可能是一个整型、实型或字符型数据,也可能是一个关系运算结果或逻辑运算结果(整型数据)。

(5)不同数据类型的数据在计算和赋值时的类型转换原则

不同数据类型的数据在计算时的类型转换原则是“就长不就短”,即先将数据长度短的数据转换成数据长度长的数据,再进行计算,计算结果为数据长度长的数据类型。

不同数据类型的数据在赋值时的类型转换原则是“就左不就右”,即将赋值运算符右边表达式的数据类型转换成左边变量的数据类型,再进行赋值。

2.2 典型例题分析与解答

1 在 C 语言中,不允许有常量的数据类型是()

- ① 整型 ② 结构型 ③ 字符型 ④ 字符串

【分析】C 语言中,允许出现整型、实型、字符型、字符串的常量,其中整型常量又区分为短整型常量和长整型常量。

【答案】②

2 下列数据中,不属于常量的是()

- ① 123L ② ' \012' ③ "12.3L" ④ 12.3L

【分析】④中的数据不是常量,因为实型常量是不区分单精度和双精度的,12.3 后跟字母 L 是错误的;①中是长整型常量;②中是以转义字符方式出现的字符型常量;③中是字符串常量。

【答案】④

3 $-12345E-3$ 代表的十进制实数是_____。

【分析】这是用指数形式表示的实型常量,对于指数形式“ $\pm aaaaaE \pm bbb$ ”的实数,其值为“ $\pm aaaaa \times 10^{\pm bbb}$ ”。

【答案】 -12.345

4 字符串“\012 \012”在内存中占用的字节数是_____个。

【分析】一个字符串所占用的内存字节数等于其中字符的数目再加 1。题目中给出的字符串中,\0 是 1 个转义字符;'0'、'1'、'2'分别都是单个字符;\012 是一个用八进制数表示的转义字符,所以其中单个字符的数目为 5,该字符串占用的内存字节数等于 $5+1$,其中增加的 1 个字节用于存放“字符串结束标记”符号 \0。

【答案】6

5 设短整型变量 x 的值为 12,假定分配给 x 的内存单元地址是 0xff00 和 0xff01,则在程序中能表示变量 x 的地址是()

- ① 0xff00 ② 0xff01 ③ &x ④ &12

【分析】C 语言规定,任何变量的地址,在程序用“& 变量名”来表示。

【答案】③

6 下列说法中,错误的是()

- ① 变量的定义可以放在所有函数之外
② 变量的定义可以不放在本编译单位中,而放在其他编译单位中
③ 变量的定义可以放在某个函数的函数头中

① 变量的定义可以放在某个复合语句的开头

【分析】①中定义的变量是正确的,这种变量是外部变量;②中定义的变量是正确的,这种变量在本编译单位中要说明为“外部参照型”变量;③中定义的变量是允许的,这种变量称为内部变量,其作用域只能是本复合语句。此外,在任何函数的函数体中都可以定义变量,所定义的变量也是内部变量,其作用域是本函数。但是,在函数中定义变量时,只能在“函数体”的开头定义,不能在“函数头部分”进行定义。

【答案】③

7 变量的存储类型主要是指_____。

【分析】定义变量时的存储类型有 4 种选择:一是“自动型(auto)”,这时变量被分配在可以重复使用的内存堆栈区;二是“寄存器型(register)”,这时变量被分配在主机(CPU)中的寄存器;三是“静态型(static)”,这时变量被分配在不能重复使用的内存数据区;四是“外部参照型(extern)”,这时仅说明该变量的定义是在其他编译单位,不在本编译单位中。由上述分析,可以看出,定义变量时选择存储类型,主要是选择给变量分配的单元在何处。

【答案】给变量分配的单元在何处

8 下列说法中,正确的是○

- ① 自动型变量是分配在内存的数据区
- ② 寄存器型变量是分配在内存的数据区中
- ③ 静态型变量是分配在内存的数据区中
- ④ 外部参照型变量是分配在内存的数据区中

【分析】按照变量在定义时的存储类型,给变量分配内存将有 3 种方式:一是内存的数据区,这个地方的单元是不能重复使用的,它指定分配给“静态型(static)”变量;二是内存的堆栈区,这个地方的单元是能重复使用的,它指定分配给“自动型(auto)”变量;三是主机(CPU)中的寄存器对,这个地方是可以重复使用的,它指定分配给“寄存器型(register)”变量;至于“外部参照型(extern)”变量不是在本编译单位中定义的,它只是用来说明需要在其他编译单位中寻找这个变量的定义,以便确定该变量的存储特性和数据类型。

【答案】③

9 从变量的存储类型来看,不能对变量进行初始化的是()

- ① extern ② auto ③ register ④ static

【分析】因为“外部参照型(extern)”变量的定义不是在这儿定义的,这儿仅是说明该变量的定义在其他编译单位中,所以这种存储类型的变量不能初始化。

【答案】①

10 下列说法中,正确的是()

- ① 从变量的作用域角度,我们把变量分为“全局变量”和“局部变量”
- ② 从不同编译单位的角度,我们把变量分为“内部变量”和“外部变量”

③ 外部变量总是全局变量,它的作用域是从定义点到本程序的末尾

④ 静态型变量总是全局变量,它的作用域是从定义点到本程序的末尾

【分析】①的分析:全局变量和局部变量是从变量的生存期角度来划分的,在程序运行期间内始终存活的变量一律是全局变量;在程序运行期间内,局部时间内存活的变量一律是局部变量。②的分析:内部变量和外部变量是从变量的定义位置来划分的,在函数内部定义的变量一律是局部变量;在函数外部定义的变量一律是外部变量。④的分析:静态型变量的存活期是整个程序的运行期,所以它是全局变量,但是,如果该变量是在函数内部定义的,那它就是内部变量,内部变量的作用域只能是所定义的函数内部,不是从定义点到程序的末尾。

【答案】③

11 表达式“ $13.5 > 13L < 2.5$ ”的值是_____。

【分析】注意这样的表达式在语法上是正确的。按照“关系运算符的优先级和结合性”可知,要先计算“ $>$ ”,后计算“ $<$ ”。计算“ $13.5 > 13L$ ”,结果为真(值为1);接着计算“ $1 < 2.5$ ”,结果为1。

【答案】1

12 设字符型变量 ch 中存放字符“A”,则执行“ $ch++ + 2$ ”后, ch 中的字符是_____。

【分析】按照多个运算符结合在一起时的确认规则:自左向右选择尽可能多的合法运算符来看,表达式“ $ch++ + 2$ ”应理解为“ $(ch++) + 2$ ”,由于“ $++$ ”是后缀运算符,所以先取出 ch 的值(字符“A”的 ASCII 代码值为 65),执行“ $65 + 2$ ”的运算,结果为 67。注意这是表达式的值,并不是变量 ch 中的值。由于变量 ch 还要进行“后缀”的“ $++$ ”运算,所以变量 ch 的值是“ $65 + 1$ ”,其值为 66,是字符“B”。

【答案】B

13 关于表达式“ $2 > 1 > 0 ? 3 > 2 > 1 : 4 > 3 > 2 ? 5 > 4 > 3 : 6 > 5 > 4$ ”的准确说法是()

① 语法出错 ② 表达式值为非 0 ③ 表达式值为 1 ④ 表达式值为 0

【分析】分析给出的表达式可知是一个条件表达式。按照条件运算符和关系运算符的优先级来说,整个表达式相当于“ $(2 > 1 > 0) ? (3 > 2 > 1) : (4 > 3 > 2) ? (5 > 4 > 3) : (6 > 5 > 4)$ ”;再从条件运算符的结合性来说,整个表达式相当于“ $(2 > 1 > 0) ? (3 > 2 > 1) : ((4 > 3 > 2) ? (5 > 4 > 3) : (6 > 5 > 4))$ ”,所以该表达式的语法是正确的。接下来我们可以具体计算这个表达式的值:首先计算局部表达式“ $(4 > 3 > 2) ? (5 > 4 > 3) : (6 > 5 > 4)$ ”, $(4 > 3 > 2)$ 的结果是 $((4 > 3) > 2) = (1 > 2) = 0$;显然这个局部表达式的值就是表达式“ $6 > 5 > 4$ ”,其值为 0。再计算整个表达式“ $(2 > 1 > 0) ? (3 > 2 > 1) : 0$ ”,其中 $(2 > 1 > 0)$ 的结果是 $((2 > 1) > 0) = (1 > 0) = 1$,所以这个表达式的值是 $(3 > 2 > 1)$,其值为 0,所以整个表达式的值为 0。

【答案】④

14 设字符型变量 ch1 中存放的字符是“A”,字符型变量 ch2 中存放的字符是“B”,则执行“ $(ch1 = 'B') || (ch2 = 'C')$ ”的运算后,变量 ch2 中的字符是_____。