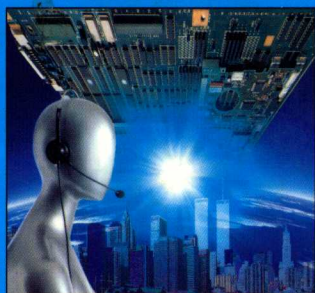
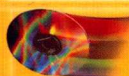


21 世纪大学教学理论与实践丛书



姚惠能 刘加海 编 著

C语言程序设计



浙江科学技术出版社

000580181

TP312
965

要目容内

21 世纪大学教学理论与实践丛书

C 语言程序设计

杨家睿

姚惠能 刘加海 编著

<http://10.12.72.101/incoming>
<http://10.12.72.103/incoming/C试卷>

10.10.2.95

图书馆



贵阳学院图书馆



GYXY650067

浙江科学技术出版社

内容提要

本书根据 C 语言的特点,力求突出系统性、实用性、简明易懂,由浅入深地讲授 C 语言的基本内容及编程特点。本书主要内容包括: C 语言基本概念、控制结构、模块化程序设计、变量的存储类型及编译预处理、指针与数组、指针与函数、结构与共用体、位运算、文件、程序设计常用数据算法简解、面向对象的程序设计概述、C++ 的基本概念、类与数据封装、继承、操作符重载、虚拟函数与多态性等。本书内容精炼,结构合理,便于自学,对读者可能遇到的难点做了十分系统、清楚和详细的阐述,极大地减轻了读者学习 C 语言的困难,是各类应试人员,包括在校大学生、学历文凭教育、自学考试学生学习的良师益友,是一本系统的应试教材。

图书在版编目(CIP)数据

C 语言程序设计/姚惠能,刘加海编著. —杭州:浙江科学技术出版社. 2000. 11

(21 世纪大学教学理论与实践丛书)

ISBN 7-5341-1353-9

I. C… II. ①姚…②刘… III. C 语言-程序设计
IV. TP312

中国版本图书馆 CIP 数据核字 (2000) 第 55909 号

丛 书 名	21 世纪大学教学理论与实践丛书
书 名	C 语言程序设计
编 著	姚惠能 刘加海
出 版	浙江科学技术出版社
印 刷	浙江印刷集团公司
发 行	浙江省新华书店
制 作	浙江科学技术出版社计算机图书工作室
读者热线	0571-5157523 5065824
电子信箱	hzzjkj@mail.hz.zj.cn
开 本	787×1092 1/16
印 张	19
字 数	470 000
版 次	2000 年 11 月第 1 版
印 次	2000 年 11 月第 1 次印刷
书 号	ISBN 7-5341-1353-9/G · 272
(全 10 册)	250.00 元 (本册 25.00 元)
总 定 价	
责任编辑	熊盛新 陈 岚
封面设计	金 晖

前言

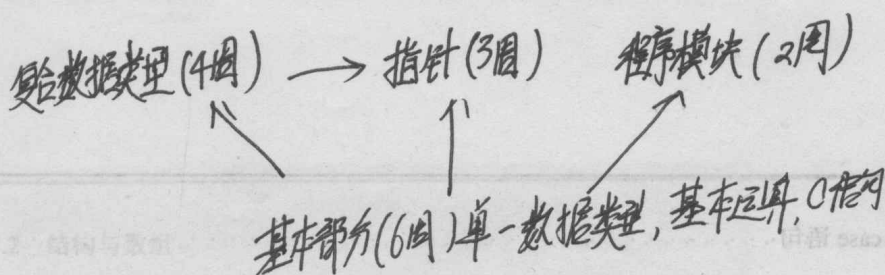
随着我国计算机应用的进一步普及和深入,学生进入高等学校时的计算机的水平比前几年有了很大的提高。作者根据多年的教学经验,在修改了在浙江大学讲授的自编讲义的基础上编写了这本《C 语言程序设计》教程。本书紧扣大纲,内容取舍得当。

在 C 语言程序设计这门课程中,学生对指针的学习及应用是本课程的重点及难点。而学生对指针的应用又无法灵活掌握,而大多数教材又把指针与数组分隔教学,使得学生对数组先入为主,造成对 C 语言的精华——指针的应用更为困难。因而作者在讲授 C 语言程序设计时,先讲指针,后讲数组,并且这些内容都在第 1 章讲授,使指针的概念及应用融合在整个教材中,达到了非常好的效果。本书共分 16 章,内容包括: C 语言基本概念、控制结构、模块化程序设计、变量的存储类型及编译预处理、指针与数组、指针与函数、结构与共用体、位运算、文件、程序设计常用数据算法简解、面向对象的程序设计概述、C++ 的基本概念、类与数据封装、继承、操作符重载、虚拟函数与多态性及附录等。本书的编写力求在体系结构上安排合理、重点突出、便于掌握;在语言叙述上注重概念清晰、逻辑性强、通俗易懂、便于自学。书中各章后均配有数量一定的习题,并且对选择题、填空题、程序阅读题给出了参考答案,以便学生自我检查。

在本书的编写和出版过程中,浙江科学技术出版社给予了大力支持,在此表示诚挚的感谢。由于编者水平有限,书中难免存在缺点和错误,真诚希望广大读者批评指正。

编著者

2000 年 11 月



目 录

C语言中大小写字母代表不同含义。(基本使用小写字母)

第1编 面向过程的程序设计

第1章 C语言基本概念	1
1.1 C语言的发展简史	1
1.2 C语言的优点及地位	2
1.3 C语言的特点	2
1.4 C语言的结构与风格	2
1.5 程序设计的一般概念	4
1.6 简单的C程序	4
1.7 关键字与标识符	4
1.8 变 量	7
1.9 整型变量	9
1.10 实型变量	11
1.11 字符变量与字符串	11
1.12 变量与地址	14
1.13 表达式	14
1.14 连续赋值运算	17
1.15 自加、自减运算	17
1.16 长度测试运算符 sizeof (类型)	18
1.17 逗号表达式	18
1.18 复合语句	19
习 题 1	19
1.19 程序的格式化输入输出函数	21
1.20 其他输入输出语句	24
1.21 地址与指针	26
1.22 数组的初步概念	32
习 题 2	35
第2章 控制结构	39
2.1 关系和逻辑运算	39
2.2 C语言的流程控制结构	41
2.3 if语句和用if语句构成的选择结构	42
2.4 if语句嵌套	43
2.5 条件运算符	44

2.6 switch case 语句	45
习 题 3	46
2.7 循环语句	48
2.8 循环嵌套	52
2.9 break 语句与 continue 语句	52
习 题 4	54
第 3 章 模块化程序设计	61
3.1 库函数	61
3.2 自定义库函数	62
3.3 函数的嵌套调用	67
3.4 函数的递归调用	67
3.5 一维数组与函数	70
3.6 二维数组与函数	71
习 题 5	71
第 4 章 变量的存储类型及编译预处理	77
4.1 局部变量、全局变量和存储分类	77
4.2 变量的类型及其作用域和生存期	77
4.3 函数的存储分类	83
4.4 编译预处理	84
4.5 文件包含处理	86
习 题 6	88
第 5 章 指针与数组	91
5.1 一维数组和指针	91
习 题 7	94
5.2 二维数组与指针	101
5.3 字符数组、字符串与指针	104
5.4 指针数组	108
5.5 多级指针	110
5.6 指向字符串的指针变量与字符数组的比较	111
5.7 用于字符串处理的函数	115
习 题 8	116
第 6 章 指针与函数	123
6.1 函数参数	123
6.2 函数指针与指针函数	125
6.3 内存分配函数	129
6.4 命令行参数	130
习 题 9	131
第 7 章 结构体与共用体	135
7.1 结构体的基本概念	135

7.2 结构与数组	136
7.3 结构与指针	136
7.4 结构与函数	139
7.5 结构嵌套	140
7.6 动态链表结构	141
7.7 共用体	147
7.8 枚举型	150
7.9 typedef 定义类型	152
习 题 10	153
第 8 章 位 运 算	163
8.1 位运算符	163
8.2 逻辑位运算符的运算功能	163
习 题 11	167
第 9 章 文 件	169
9.1 文件的概念	169
9.2 有关文件的操作	171
习 题 12	181
第 10 章 程序设计常用数据算法简解	191
10.1 若干初等数学问题	191
10.2 常用数值问题算法	196
10.3 常见排序算法	200
习 题 13	204

第 2 编 面向对象的程序设计概述

第 11 章 面向对象程序设计的基本思想和有关概念	206
11.1 面向对象抽象的原理	206
11.2 面向对象计算的基本特征	207
11.3 C++的特点	207
11.4 C++对 C 语言作了很多改进	208
习 题 14	209
第 12 章 C++的基本概念	210
12.1 C++注释	210
12.2 C++的输入输出函数	210
12.3 变量说明的位置	211
12.4 内联函数	212
12.5 缺省函数参数	213

12.6	const 修饰符	214
12.7	void	215
12.8	域分辨操作符::	215
12.9	引用类型	216
12.10	运算符 new 和 delete	218
12.11	函数重载	220
习 题 15		220
第 13 章	类与数据封装	223
13.1	类定义	223
13.2	对象的定义	225
13.3	数据封装	226
习 题 16		240
第 14 章	继 承	243
习 题 17		252
第 15 章	操作符重载	254
习 题 18		259
第 16 章	虚拟函数与多态性	261
习 题 19		272
附 录		277
附录 1	常用字符与 ASCII 码对照表	277
附录 2	双目算术运算中两边运算量类型转换规律	277
附录 3	运算符及其优先级汇总表	278
附录 4	C 语言部分常用库函数	278
附录 5	Turbo C 2.0 上机操作指南	283
附录 6	BC++环境	286
附录 7	VC++编程环境中的各项命令	287
附录 8	部分习题答案	289
附录 9	参考文献	294

第1编 面向过程的程序设计

第1章 C语言基本概念

1.1 C语言的发展简史

当今，计算机已广泛应用于社会生活的各个领域，成为大众化的现代工具。计算机是一种具有内部存储能力、由程序自动控制的电子设备。人们将需要计算机做的工作写成一定形式的指令，并把它们存储在计算机的内部存储器中，当人们给出命令之后，它就按指令操作顺序自动进行。人们把这种可以连续执行的一条条指令的集合称为“程序”。可以说，程序就是人与机器进行“对话”的语言，也就是我们常说的“程序设计语言”。

目前，在社会上使用的计算机程序设计语言有上百种，它们都被称为计算机的“高级语言”，如 Vasiul Basic 语言、Pascal 语言以及本书将介绍的 C 语言等。这些语言都是用接近人们习惯的自然语言和数学语言作为语言的表达形式，人们学习和操作起来感到十分方便。

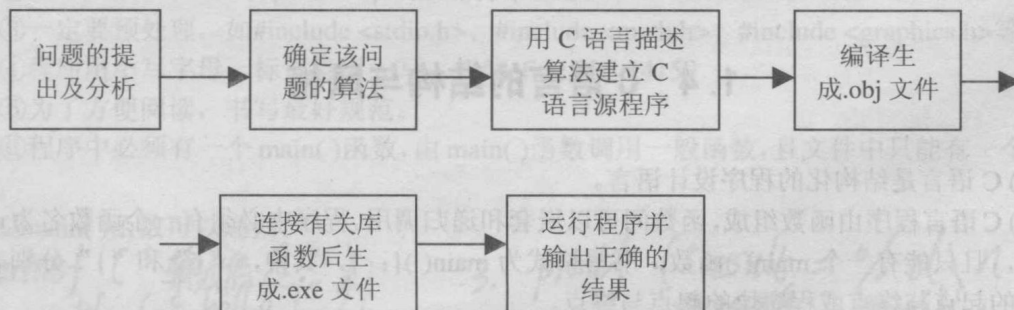
在各种计算机语言中，C 语言可以说是较为成功的。C 语言是在 UNIX 操作系统的研制和开发过程中诞生的。1997 年，美国电话与电气公司（AT &T）的贝尔实验室中的丹斯·里奇（Dennis Ritchie）先生，在 B 语言的基础上写成了 C 语言。1972 年投入了使用，他同肯·汤普森（Ken Thompson）于 1973 年用 C 语言重写了 UNIX 操作系统，并且成了以后 UNIX 各种版本的发展基础。

对于计算机本身来说，它并不能直接识别由高级语言编写的程序，它只能接受和处理由 0 和 1 的代码构成的二进制指令或数据。由于这种形式的指令是面向机器的，因此也称为“机器语言”。

我们把由高级语言编写的程序称为“源程序”，把由二进制代码表示的程序称为“目标程序”。如何把源程序转换成机器能够接受的目标程序，软件工作者编制了一系列软件，通过这些软件可以把用户按规定语法写出的语句一一翻译成二进制的机器指令。这种具有翻译功能的软件称为“编译程序”。每一种高级语言都有与它对应的编译程序。

我们所写的每条 C 语句组成的序列称为 C 源程序，它的后缀为 .c，C 源程序经过编译（Compile）后生成一个后缀为 .obj 的二进制文件，最后由“连接程序”（link）把此 .obj 文件与 C 语言提供的各种库函数连接起来生成一个 .exe 文件。在 DOS 状态下该文件就可运行。

程序的设计过程为：



1.2 C 语言的优点及地位

现在人们广泛地采用 C 语言来完成各种编程任务，是因它具有下列独特优点：

- (1) C 语言是一种通用的结构化程序设计语言，用 C 编写的程序由多个模块（函数）组成，每个模块完成一项功能。
- (2) C 语言的语句简洁、灵活，使用起来得心应手，它一共只有 32 个关键字，便于初学者掌握。
- (3) C 语言具有丰富的运算符，使用起来简单精炼，生成的机器代码质量高，内存占用少，运行速度快。
- (4) C 语言的数据类型丰富，可进行复杂的数据结构运算，它的指针数据类型的使用更有特点，它使参数传递简单、迅速，节省内存。
- (5) C 语言具有位（bit）操作能力，可直接对计算机硬件的物理地址进行访问，它既是高级语言又兼有低级语言的功能。
- (6) 灵活、移植性好、应用面广。

一般来说，汇编语言处于硬件与系统之间，而数据库语言 Cobol 处于用户与软件之间，Pascal 语言更适合于系统与软件之间，C 语言偏向于硬件与系统方面，并且能完成其他高级语言能完成的任何工作。因而，更确切地说，C 语言是一种中级语言。

1.3 C 语言的特点

- (1) 具有汇编语言的精华，具有强大外设能力，引入结构、指针、地址和位运算，可直接访问硬件，可对位、地址进行操作。
- (2) 是一种结构化程序语言，即顺序、选择（if—else、switch—case）和循环（for、do while）这 3 种结构，易于理解，便于软件的维护。
- (3) 是一种模块化结构的程序设计语言。C 程序是函数结构，模块间可以相互调用，为集体开发软件工程技术提供方便。
- (4) 具有丰富的运算符。如 +、-、×、/、&&、||、!、&、|、~、^、>>、++、-- 等。
- (5) 可使用 #define、#include 等编译预处理语句、宏定义、条件编译。
- (6) 丰富的类型，如整型、实型、双精度、字符、数组、指针、结构、共用体，具有很强的数据处理能力。
- (7) 较高的移植性，描述准确，目标程序高，语言紧凑，存储空间小。

1.4 C 语言的结构与风格

- (1) C 语言是结构化的程序设计语言。
- (2) C 语言程序由函数组成，函数间可以嵌套和递归调用。程序中必须有一个函数名为 main() 的函数，且只能有一个 main() 函数，书写格式为 main(){; }，其中，“{”和“}”分别表示程序执行的起点与终点或程序块的起点与终点。

(3) C 程序可由多个文件组成。
(4) C 语言程序书写格式自由，一行内可写几个语句，一个语句也可写在多行上，C 语言程序没有行号。C 语言的结构如下图所示。

文件预处理	
函数名（形式参数）	
函数体	数据定义部分
	执行语句部分

格式控制
"%..." d—输出十进制整数
f—输出...浮点数
c—输出单个字符
转义字符 \n, \t.

例题分析:

```
#include <stdio.h> /*文件预处理*/
int add(int x, int y) /*函数名*/
{ /*函数体开始*/
    int z; /*数据定义部分*/
    z=x+y; /*执行语句部分*/
    return z;
} /*函数体结束*/
```

格式化输入函数 scanf()
—scanf (格式控制, 输入参数1,..., 输入参数n);
scanf ("%d %d", &m, &n);

在程序中可以对程序进行注释，注释部分必须用符号“/*”和“*/”括起来。“/”和“*”之间不可以有空格。注释可以用西文，也可以用中文。注释可以出现在程序中任意合适的地方。注释部分对程序的运行不起作用。在注释中可以说明变量的含义、程序段的功能，以便帮助人们阅读程序。因此一个好的程序应该有详细的注释。

在上例中，如要在机器上运行，必须要有一个 main() 函数，通过 main() 调用函数 add()，例如：

```
main( )
{ int a=1, b=2, z;
  z=add(a, b);
  printf("%d+%d=%d\n", a, b, z);
}
```

对于普通变量一定需要地址符。用回车表输入结束
0 字符输入函数 getcharr()
例: char ch;
ch=getchar();
0 字符输出函数 putchar

此程序的执行结果为: 1+2=3

请读者仔细思考，有关问题将在以后学到。

良好的编程风格有助于对程序的阅读、理解和记忆，因而编程时应养成良好的习惯，并注意以下几点：

- ①用“;”作为语句的结束记号，但循环体、main()、#include<>、#define 不是语句，因而后面不能用“;”号。例如：
int i;
for(i=1; i<=5; i++) /*此位置无“;”号*/
printf("Hello\n");
- ②“/*”和“*/”，“{”和“}”必须成对使用。
- ③一定要预处理，如#include <stdio.h>、#include <math.h>、#include <graphics.h>等。
- ④程序用小写字母，标识符用大写，如#define PI 3.14159
- ⑤为了方便阅读，书写最好规范。
- ⑥程序中必须有一个 main() 函数，由 main() 函数调用一般函数，且文件中只能有一个 main() 函数。

⑦main() 函数可以递归。

```
1. printf ("输出信息");  
   printf ("hello!");  
3. printf ("result = %d\n", resm);  
   result = m!
```

1.5 程序设计的一般概念

目前的程序设计类型一般可分为非结构化程序设计和结构化程序设计，以及面向对象的设计，C 语言是结构化程序设计。

在程序设计的过程中一般包括：

- (1) 问题的提出、要求及所采用的数据结构。
- (2) 算法的确定、程序的编制。
- (3) 程序的调试及修改。
- (4) 整理并写出文档资料。

结构化程序设计由 3 种结构组成：顺序结构、选择结构和循环结构。

1.6 简单的 C 程序

例题分析：

```
/*最简单的 C 程序*/
```

```
#include <stdio.h>
```

```
main()
```

```
{  
;  
}
```

此程序没有什么意义，执行后什么也没做。

例题分析：

```
/*功能：打印字符，在屏幕上输出：Hello, world */
```

```
#include <stdio.h>
```

```
main()
```

```
{  
    printf("Hello, world\n");    /*\n 为换行符*/  
}
```

此程序执行后在屏幕上输出：

Hello, world

1.7 关键字与标识符

在 C 语言中，标识符可用作变量名、符号名、函数名和后面将要学到的数组名、文件名以及一些具有专门含义的名字。合法的标识符由字母、数字和下划线组成，并且第一个字符必须为字母或下划线，不能跨行书写，自定义标识符不能与关键字同名。下面的标识符都是合法的：

are、PI、liu_123、liu、e_array

以下都是非法的标识符：

456P、cade-y、w.w、a&b

在 C 语言的标识符中，大写字母和小写字母被认为是两个不同的字符，因此 page 和 Page 是两个不同的标识符。

对于标识符的长度（即一个标识符允许的字符个数），一般的计算机系统规定取前 8 个字

符有效,如果多于8个字符,多余的字符将不被识别。如 num12345 和 num123456 在计算机系统内则被认为是相同的标识符,即 num12345。有些系统允许取较长的名字,读者在取名时应当了解所用系统的具体规定。

C 语言的标识符可以分为以下 3 类:

一、关键字

关键字表示某种特定含义的英文单词,如 include、it、char、for 的含义特定,不允许随意书写或拼写错误。它们不能再用作变量名或函数名。

C 语言中的关键字有 continue、else if、typedef、break、default、for、return、struct、case、do、goto、switch、union、while、char、auto、unsigned、double、extern、sizeof、float、register、void、int、static、long、short。

二、预定义标识符

有些标识符在 C 语言中也都有特定的含义,如 C 语言提供的库函数的名字(如 printf)和预编译处理命令(如 define)等。C 语言语法允许用户把这类标识符另作它用,但这将使这些标识符失去系统规定的原意。鉴于目前各种计算机系统的 C 语言都一致把这类标识符作为固定的库函数名或预编译处理中的专门命令使用,因此为了避免误解,建议用户不要把这些预定义标识符另作它用。

三、用户标识符

由用户根据需要定义的标识符称为用户标识符。一般用来给变量、函数、数组或文件等命名。程序中使用的用户标识符除要遵循起名规则外,还应注意做到“见名知义”,即采用具有相关含义的英文单词或汉语拼音,以增加程序的可读性。

如果用户标识符与关键字相同,程序在编译时将给出出错信息;如果与预定义标识符相同,系统并不报错,只是该预定义标识符将失去原定含义,代之以用户确认的含义;或者会引发一些运行时的错误。

例题分析: 下列各组字符序列中,可用作 C 语言程序标识符的一组字符序列是()。

- (A) S.b, sum, auerage, _above (B) Class, day, lotus_1, 2day
(C) #md, &l2x, month, student_n1 (D) D56, r_1_2, name, _st_1

解析: C 语言规定标识符只能由字母、数字和下划线 3 种字符组成,而且第一个字符必须是字母或下划线,因而答案为 D。

例题分析: 下列字符序列中,不可用作 C 语言标识符的是()。

- (A) b70 (B) #ab (C) _symbo (D) a_1

解析: C 语言规定标识符只能由字母、数字和下划线 3 种字符组成,而且第一个字符必须是字母或下划线,因而答案为 B。

四、常量

在程序运行过程中,其值不能被改变的量称为常量。常量通常分为普通常量与无名常量及符号常量。在 C 语言中,常量有不同的类型,有整型常量(int)、实型常量(float 或 double)、字符常量(char)和字符串常量。即使是整型常量,也还有短整型(short int)、长整型(long int)和无符号型(unsigned int)。

(一) 普通常量

实型常量也称为数值型常量,它们有正值和负值之分。基本整型常量只用数字表示,不能带小数点,如 12、-1、0 等。实型常量通常用带小数点的数表示,如 3.14159、2.71828、0.0、.54 等。‘A’则是字符型常量。

由此可见,常量的类型从字面形式即可区分是整型常量、字符型常量还是实型常量。C 编译程序就是以此来确定数值常量的类型的。

可以用一个符号名来代表一个常量，但是这个符号名必须在程序中进行特别的“指定”，这在程序中叫做定义变量，即给变量分配存储空间。

- 实型常量定义，如 `float x=2.1;`
- 整型常量定义，如 `int y=4;`
- 字符常量定义，如 `char ch='A';`

(二) 无名常量

在 C 语言中，整型常量可以用十进制、八进制和十六进制格式表示。十进制常量用一串连续的数字来表示，如 32767、-32768、0 等。八进制数用数字 0 开头（注意不是字母）。例如，05、012、01 都是八进制数，它们分别代表十进制数 5、10、1。因此，在 C 程序中不能在一个十进制整数前面加前导零。注意，八进制数只能用合法的八进制数字表示。例如，不能写成 018，因为数字 8 不是八进制数字。十六进制数用数字 0 和字母 x（或大写字母 X）开头。例如，0x10、0xff、0x8 均是十六进制数，它们分别代表十进制数 16、165、8。注意：十六进制数只能用合法的十六进制数字表示，字母 a、b、c、d、e、f 既可用大写字母也可用小写字母。

例：十进制常量表示方法：123

十六进制常量表示方法：0x123

八进制常量表示方法：0123

注意：常量的长度及范围与机器类型有关。

例：16 位机整数：-32768~32767

32 位机整数：-2147483648~2147483647

例题分析：

(1) 在 C 语言中，错误的 int 类型的常数是()

(A) 32768

(B) 0

(C) 037

(D) 0xAF

解析：int 型的范围是 -32768~32767，32768 已超出 int 型数据的范围，C 为 8 进制，D 为 16 进制，所以答案为 A。

(2) 下列常数中不能作为 C 语言常量的是()

(A) 0xA5

(B) 2.5e-2

(C) 57

(D) 0582

解析：D 为 8 进制表示形式，但 D 包含了数字 8，所以答案为 D。

(3) 下列形式的常数中，C 程序不允许出现的是()。

(A) 45

(B) ±123

(C) 25.6e-2

(D) 4e3

解析：C 程序允许出现的常数为有一个确定值的整数、实数（可用小数形式或指数形式）等。±123 不允许出现。答案为 B。

(三) 符号常量

定义格式：

#define <常量名><常量值>

例：**#define PI 3.14159**

它的含义是指凡是在程序中出现常量名 PI 的地方，都用常量值 3.14159 来替换。

注意：①在程序中替换时，不作语法检查。

②常量名、常量值之间无等号，常量值后无分号。

③常量名一般大写。

例题分析：符号常量在程序中的应用：计算圆面积。

```
#include <stdio.h>
```

```
#define PI 3.14159 /*定义符号名 PI 为 3.14159*/
```

```
main()
```

```
{
```

%u 表示无符号数

```
float s, r=5.0;
s=PI*r*r;
printf("s=%f\n", s); }
```

main()

1.8 变量

{ int x=40000
printf("%d", x); }

输出: -25536

变量是指在程序运行过程中其值可以发生变化的量，通常用来保存程序运行过程中的输入数据、计算获得的中间结果和最终结果。

变量的取名规则同标识符，其中的英文字母常用小写。C语言系统本身也使用变量名，一般都是以下划线（_）开头的，为了区别，用户程序中的变量名一般不以下划线开头。给变量取名时，为了便于阅读和理解程序，一般都用代表变量值或用途的标识符。可以是英文单词或缩写，也可以是中文拼音字母或缩写。

当程序运行时，每个变量都要占用连续的若干字节，所占用的字节数由变量的数据类型确定。其中第一个字节的地址称为变量的地址。C语言规定，程序中变量的地址是用“&变量名”来表示的。

一、量的数据类型及其定义

C语言规定，变量可以是任何一种数据类型，如整型、短整型、长整型、无符号整型、无符号短整型、无符号长整型、单精度型、双精度型、字符型等基本类型，或者是数组型、结构型、共用型等构造类型，也可以是指针型、枚举型。

通常把具有某种数据类型的变量叫做该类型变量。例如，短整型变量、长整型变量、单精度型变量、双精度型变量、字符型变量，等等。

需要注意的是，字符串只能是常量，C语言中没有字符串变量。

每个变量在使用前都必须定义，定义的内容之一就是说明其数据类型。

定义变量数据类型的语句格式如下：

数据类型符变量名1，变量名2，…；

数据类型符及其含义见表1.1。

表 1.1 基本数据类型符

数据类型	数据类型符	占用字节数	数值范围
整型	int	2	-32768~+32767
短整型	short	2	-32768~+32767
长整型	long	4	-2147483648~+2147483647
无符号整型	unsigned int	2	0~65535
无符号短整型	unsigned short	2	0~65535
无符号长整型	unsigned long	4	0~4294967295
单精度实型	float	4	-10 ³⁸ ~10 ³⁸
双精度实型	double	8	-10 ³⁰⁸ ~10 ³⁰⁸
字符型	char	1	-128~+127

表中，整型和无符号整型所占用的字节数是随计算机的不同而不同的，在大部分计算机上都是和短整型占用相同的字节数（两个字节）。

对变量进行定义时，要注意下列几点：

- ①对变量的定义可以放在函数之外，也可以放在函数体或复合语句中。如果是放在函数体或复合语句中，则必须集中放在最前面。
- ②被定义为整型（包括 int、short、long）的变量，若其值在-128~127 之间，可以当做字符

型变量使用；被定义为无符号整型（包括 unsigned int、unsigned short、unsigned long）的变量，若其值在 0~255 之间，也可以当做字符型变量使用。被定义为字符型的变量，可以当做整型（包括 int、short、long）变量使用，其值将在-128~127 之间，也可以当做无符号整型（包括 unsigned int、unsigned short、unsigned long）变量使用，其值在 0~255 之间。

二、变量的存储类型及其定义

当我们说明了一个变量的数据类型，C 语言编译系统就要给该变量安排若干个字节，用来存放该变量的值。我们知道，在计算机的寄存器和内存中都可以存放数据，而内存中又可以分为一般数据区和堆栈区。我们把变量存放在何处称为变量的存储类型。用户可以通过说明变量的存储类型来选择变量的具体存储地点。

定义变量存储类型的语句格式如下：

存储类型符 数据类型符 变量名 1，变量名 2，…；

存储类型符及其含义见表 1.2。

表 1.2 存储类型符表

存储类型	存储类型符	存储地点
自动型	auto	内存堆栈区
寄存器型	register	CPU 的通用寄存器
静态型	static	内存数据区
外部参照型	extern	

（一）自动型

自动型又称堆栈型。自动型变量被分配在内存的堆栈区，堆栈区内存在程序运行中是重复使用的。当某个函数中定义了自动型变量，C 语言就在堆栈区给该变量分配字节用于存放变量的值。当退出该函数时，C 语言就释放该变量，即从堆栈区中收回分配给该变量的字节，以便重新分配给其他自动型变量。这样做的目的是节省内存。

用户对于只在某函数中使用的变量应该将其定义成自动型变量，以便充分利用内存。当定义某个或某些变量时，省略了存储类型符，系统将默认为自动型变量。

（二）寄存器型

寄存器型变量分配在 CPU 的通用寄存器中。由于 CPU 具有的通用寄存器数量有限，C 程序中允许定义的寄存器型变量一般以两个左右为宜。如果定义为寄存器型变量的数目超过所提供的寄存器数目，编译系统自动将超出的变量设为自动型变量。对于占用字节数多的变量，如 long、float、double 类型的变量，不能定义为寄存器型变量。

寄存器型变量一般是在函数中定义的，退出该函数后就释放它所占用的寄存器。

（三）静态型

静态型变量是分配在内存的数据区中，它们在程序开始运行时就分配了固定的字节，在程序运行过程中不释放。只有程序运行结束后，才释放所占用的内存。

（四）外部参照型

C 语言允许将一个源程序清单分放在若干个程序文件中，采用分块编译方法编译生成一个目标程序。其中每个程序文件称为一个“编译单位”。

外部参照型变量是专用于多个编译单位之间传递数据用的。当编译单位甲中要使用在编译单位乙中定义的变量，则编译单位甲就要说明该变量是外部参照型，以便 C 语言编译系统在编译单位甲之外的其他编译单位中寻找该变量的定义。将某个变量说明为其他编译单位定义的变量的方法就是将它说明为“外部参照型”。被说明成外部参照型的变量的存储类型和数据类型都由定义该变量的那个编译单位来说明。

1.9 整型变量

整型变量可以分为基本型 `int`、短整型 `short int`、长整型 `long int` 和无符号型 `unsigned` 这4种。本节只介绍基本类型的整型变量。

基本型的整型变量在内存中占两个字节，用类型名关键字 `int` 进行定义，定义形式如下：

①定义：`int <变量名 1>`;

`int <变量名 2>`;

或 `int <变量名 1>, <变量名 2>.....<变量名 n>`;

例：`int k;` /*定义变量 `k` 为整型变量*/

②一个定义语句必须以一个“;”号结束。在一个定义语句中也可以同时定义多个变量，变量之间用逗号隔开。如：

`int i, j, k;` /*定义变量 `i`、`j`、`k` 为整型*/

一般微机为基本型整型变量开辟两个字节（16个二进制位）的内存单元，并按整型数的存储方式存放数据。整型的变量只能存放整型数值。基本型整型变量中允许存放的数值范围是-32768~32767。

当在程序中用以上方式定义变量 `i`、`j` 和 `k` 时，编译程序为变量 `i`、`j` 和 `k` 都开辟存储单元，但是并没有在存储单元中放置任何初值，因此在这些存储单元中，原有的信息并没有被清除，这时变量中的值是无意义的，称变量“无定义”。

C语言规定，可以在定义变量后再给变量赋值。

如：`int x, k;` /*此处定义 `x`、`k` 为整型变量*/

`x=2, k=3;` /*定义变量后才可给变量赋值*/

也可把这两个语句合成一个，定义的同时给变量赋值，这叫做变量的初始化，例如：

```
#include <stdio.h>
```

```
main()
```

```
{ int x=2, k=3; /*定义 x、k 为整型变量，它们的初值分别为 2、3*/
```

```
}
```

在前两节中已经介绍过 `int` 类型的整型常量和整型变量。通常称 `int` 类型为基本整型。

除此以外，C语言中还包括其他3种整数类型，它们是短整型（`shortint`）、长整型（`long int`）、无符号型（`unsigned`）。若不指定为无符号型，隐含的即为有符号型（`signed`），无论是短整型还是长整型数，都被识别为有符号整数。不同的计算机系统对这几类整型数所占用的字节数和数值范围有不同规定。

长整型数的表示方法：在一个整型常量后面加一个字母后缀 `l`（`L` 的小写）或 `L`，如 `123L`、`345l`、`0L` 等，这些常量在内存中占4个字节。注意：一个数，如在数字后面未加后缀字母 `L`，也不能表示一个长整型数。

无符号整数的表示方法：在数字的末尾应该加字母后缀 `u` 或 `U`，若是长整型无符号整数常量，则应该加后缀 `lu` 或 `LU`；短整型无符号常量取值应在 0~65535 范围内，长整型无符号常量的取值则在 0~4294967295 范围内。

注意：无符号常量不能表示成小于0的负数，如 `-200U` 是不合法的。

计算机中，内存储器的最小存储单位称为“位（`bite`）”，每一位存放数字0或1，因此称为二进制位。大多数计算机把8个二进制位组成一个“字节（`byte`）”，并给4个字节分配一个地址。若干字节组成一个“字（`word`）”，用一个字来存放一条机器指令或一个数据。一个字