

经典

C面试 真题精讲

李亚锋 编著

深入研究Google、Intel、Microsoft、IBM、华为、中兴、
百度等上百家国际知名IT企业的C语言面试、笔试题

帮助读者掌握企业要求应聘人员具备的C语言核心技术，
掌握C语言在实际项目开发中的应用知识

清华大学出版社

经典
C面试
真题精讲

李亚锋 编著

清华大学出版社
北京

内 容 简 介

C 语言是计算机编程史上生命力最强、影响力最广的一种编程语言,迄今为止 C 语言仍旧是开发领域中最重要、最受欢迎的编程语言之一。本书以 Google、Intel、Microsoft、IBM、华为、中兴、百度等上百家国际知名 IT 企业的 C 语言面试、笔试题为研究对象,深入分析这些典型 C 面试、笔试题的出题意图及解题方法。本书不仅仅是为了让读者了解知名企业的面试、笔试题目,更重要的是让读者明白企业需要应聘人员具备哪些 C 语言核心技术,以及让读者明白在实际项目开发中 C 语言是如何被灵活运用的。这是一本非常适合求职人员在求职路上征战参考的技术宝典,也是一本值得各大企业招聘使用的参考手册。

本书不仅适合高校软件工程专业的本科和研究生学习使用,也适合在职软件工程师工作参考之用。

本书封面贴有清华大学出版社防伪标签,无标签者不得销售。

版权所有,侵权必究。侵权举报电话:010-62782989 13701121933

图书在版编目(CIP)数据

经典 C 面试真题精讲 / 李亚锋编著. —北京:清华大学出版社, 2011.10

ISBN 978-7-302-25946-6

I. ①经… II. ①李… III. ①C 语言—程序设计—水平考试—题解 IV. ①TP312-44

中国版本图书馆 CIP 数据核字(2011)第 122813 号

责任编辑:夏兆彦

责任校对:徐俊伟

责任印制:何 芊

出版发行:清华大学出版社

地 址:北京清华大学学研大厦 A 座

<http://www.tup.com.cn>

邮 编:100084

社 总 机:010-62770175

邮 购:010-62786544

投稿与读者服务:010-62795954, jsjic@tup.tsinghua.edu.cn

质 量 反 馈:010-62772015, zhiliang@tup.tsinghua.edu.cn

印 装 者:三河市金元印装有限公司

经 销:全国新华书店

开 本:185×260 印 张:14.75 字 数:372 千字

版 次:2011 年 10 月第 1 版 印 次:2011 年 10 月第 1 次印刷

印 一 数:1~4000

定 价:39.00 元



李亚锋： 陕西人，曾就职于威盛电子、O2Micro、WYSE等公司，有8年以上嵌入式系统开发经验，专注于嵌入式Linux系统的设备驱动与应用开发。主要负责网络设备、消费类电子设备的开发工作，对标准C语言编程有丰富的项目开发经验。其著作有：《ARM嵌入式Linux系统开发从入门到精通》（清华大学出版社）和《ARM嵌入式Linux设备驱动实例开发》（中国电力出版社）。

C 语言在计算机发展历史上是一个古老的话题，它不仅对操作系统的发展起到了推波助澜的作用，也对整个计算机产业的长久发展有着重大意义。对于我自己来说，C 语言是我编程语言中的母语，我从学习和使用 C 语言编程到现在已经超过十年。十年对于每个人的职业生涯来说就不是一个短暂的周期，并且 C 语言一直是我工作中项目开发的优选编程语言，即使现在也不例外。

在当今互联网成为人们生活必备品的时代，“码农”成为了当今 IT 界最流行词之一。码农，顾名思义就是写代码的农民，更多的是一种嘲讽的叫法，“码农”一词不管是用来自嘲还是他嘲，我认为最重要的还是编程人员自身的心态。曾经连续 13 年蝉联世界首富的比尔·盖茨，Google 创始人拉里·佩奇和谢尔盖·布林，苹果公司 CEO 史蒂夫·乔布斯和百度公司 CEO 李彦宏等，他们当年也是“码农”。很多人只看到了他们成功后的光环，却忽略了他们的不懈努力。对于“码农”的称号，我的理解就是要像农民耕地一样去写代码，只有踏踏实实、勤勤恳恳，你的生命才会虚度，你也才有机会实现自己的理想。既然选择了“码农”职业，那么就要做一个合格的、优秀的“码农”。

最近一年时间里，笔者利用了几乎所有的业余时间，经历了两百多个夜晚和几十个休息日的不懈奋战，使这本书终于与大家见面了。

为什么要写这本书？

首先，应一位在出版社工作的朋友的真诚邀约，盛情难却决定再写一本书。笔者曾编写过两本书，深知写书的不易，也明白期望通过稿费带来的收入是非常有限的。在国内很少能看到技术大牛的专著，因为在这个市场经济的大环境中，写技术类书籍意味着做慈善事业，因为需要花费大量的精力和时间，毕竟不是每个人都能成为大慈善家，读者也应该可以理解。

其次，C 语言书籍市面上目前已有成百上千种，为什么笔者还要选择这个方向？笔者认为这个方向是有市场的。尽管 C 语言书籍种类繁多，但真正面向真实企业面试和项目开发的书籍并不多见。本书的主要目标就是帮助读者有针对性地学习，不仅仅是为了帮助读者轻松通过技术面试，更重要的是让读者明白在实际开发中，这些知识点是如何被灵活运用到的。

再次，谭浩强老师的 C 语言书籍是国内众所周知的大学教程，该系列书籍对 C 语言语法的讲解比较全面，笔者也是从谭老师的书籍开始学习 C 语言的。如果让笔者非要说谭老师书籍的不足，那就是缺少与实际编程的过渡性，没有重点突出 C 语言在实际编程中的灵活运用技巧。提起 C 语言经典书籍，很多人都会提到《The C Programming Language》、《Expert C Programming》、《Pointers on C》和《C Traps and Pitfalls》等，的

确这些书籍在全世界都被公认为经典的 C 书籍，可惜对英语基础不好的读者来说，很难掌握书中的内容。鉴于此，笔者决定写这本书，希望它更适合国内读者学习。

最后，笔者从工作至今一直从事 C 语言方面的项目开发，对其有一定的认识和理解，想借这本书做一个总结与巩固，同时也希望对 C 语言的普及和推广做一些力所能及的帮助。

本书组成

第 0 章 笔试题自测

在学习本书之前，笔者建议读者先感受一下企业招聘时的部分真实 C 笔试、面试题，这不仅仅是一个自然测验，更是下一步学习的目标、导航。

第 1 章 灵魂——指针

C 语言编程的精髓就是对指针的精巧使用，利用指针变量可以表示各种类型的数据结构，并能像汇编语言一样处理内存地址，可以编出精练而高效的程序。指针极大地丰富了 C 语言的功能，把指针比作 C 语言的灵魂一点都不为过。本章重点讲述指针与字符串、数组、结构体、联合体等常见笔试题。要想学习指针的灵活运用，建议读者阅读本章内容。

第 2 章 数据存储——内存

内存是计算机中重要的部件之一，它是与 CPU 进行沟通的桥梁。C 语言允许直接访问物理地址，能实现汇编语言的大部分功能，可以直接对硬件进行操作。通常 CPU 以字节为单位对内存进行访问，为每个存储单元顺序编号，这个编号就被称为内存地址。使用 C 语言可以方便地实现对内存的访问（读或写操作），非常适合开发系统软件。在写程序过程中时刻要提醒自己“正确、有效地使用内存”。

第 3 章 程序的思想——算法

如果把计算机程序比作一个人的话，那么程序算法就是人的大脑，它是程序的设计思想。程序算法是对特定问题求解过程的描述，是指令的有限序列，每条指令完成一个或多个操作。通俗地讲，算法就是为解决某一特定问题而采取的具体有限的操作步骤。要想学习算法在程序中如何被应用，建议读者阅读本章内容。

第 4 章 程序的基本单位——函数

函数是 C 程序的基本组成部分，也是最重要的组成部分，可以说没有函数就没有 C 程序。一个优秀的 C 程序开发人员最关键的技能就是能在有限的时间内写出高质量、高效率的函数。本章重点讲述函数的编写原则以及调用方式。笔者更多的是站在项目开发与企业招聘的角度去讲述，所有的知识点都来源于实际的项目经验与知名企业的笔试要求，适合所有 C 语言学习者借鉴参考。

第5章 数据的传递——网络

当今提到“网络”一词，一般都会想到互联网（internet），互联网改变了人们的生活，让世界真的变成了“地球村”，当今生活没有了网络就如同没有了水、电一样。在计算机领域中，网络就是用物理链路将各个孤立的工作站或主机相连在一起，组成数据链路，且以功能完善的网络软件（网络协议、信息交换方式及网络操作系统等）实现网络资源共享的系统，从而达到资源共享和通信的目的。在众多 IT 企业中，有不少公司都从事着与网络相关的研发业务，比如通信终端、通信协议、网络应用软件等。本章将学习 C 语言在网络开发中的应用。

第6章 专用的计算机系统——嵌入式

嵌入式技术具体会应用在哪些产品上呢？例如，消费类电子产品（手机、数码相机、MP3/MP4 等）、网络终端（路由器、交换机、硬件防火墙等）、医疗器械、机器人等。可见嵌入式产品随处可见，与人类生活密不可分，企业对嵌入式人才需求旺盛，这也是笔者单独列举此章的缘故。也许你感觉嵌入式 C 编程很奇妙，其实嵌入式 C 编程和普通的 C 编程并没有区别，因为两者都是使用 C 语言的编程语法规则，嵌入式 C 编程只是在编程时更多的要考虑资源的有限性、程序的可移植性等条件，并且应用到了 C 语言的一些特殊的功能而已。本章重点学习 C 语言在嵌入式产品开发中的核心应用。

第7章 终结语

本章首先推荐给读者一些良好的编程习惯，它会使读者写出质量较高的代码，减少出错。其次，本章分享笔者曾在编程中遇到的问题，通过这些问题的分析可使读者避免发生类似的错误。最后，笔者推荐给读者一些学习 C 语言的实用技巧。

致谢

本书的完成离不开家人的支持，在此特别感谢妻子默默无闻的奉献，是她承得了众多家务才使得我可以安心地写书。其次，由于写书占据了大部分的休息时间，使得我陪孩子的时间变得非常稀少，在此也感谢我的儿子。有付出才会有回报，正因为他们和我的付出，才有了这本书与大家见面的机会。

为了能与读者更方便地交流，读者可以访问作者的 blog:<http://mikelee.blog.51cto.com/>，笔者愿意与更多的读者成为朋友，一起交流技术。由于笔者水平有限，对本书的不足之处欢迎各位读者、专家提出宝贵意见。

小麦

2011 年 3 月

第0章 笔试题自测 1

第1章 灵魂——指针 5

1.1 指针与字符串 5

1.1.1 知识点题型 5

101 p++ 5

102 打印内存地址 7

103 sprintf()与 snprintf() 7

104 指针常量与常量指针 8

105 字符与字符串 9

106 指针变量与普通变量 10

107 指针运算 11

108 “%-10.5s” 11

1.1.2 综合题型 12

109 字符串复制 12

110 统计字符个数 14

1.2 指针与数组 16

1.2.1 知识点题型 16

111 一维数组与指针 16

112 数组中 strlen 与 sizeof
的区别 16

113 多维数组 17

114 数组作为参数传递给
函数 19115 char (*(x())[5])()与
typedef 19

1.2.2 综合题型 20

116 编程求较大数的
阶乘 20

117 字符旋转 22

1.2.3 数组总结 25

1.3 指针与结构体 26

1.3.1 知识点题型 26

118 __attribute__((packed))
含义 26119 union 与 struct
类型区别 28

1.3.2 综合题型 29

120 Little-endian 与
Big-endian 29

1.3.3 结构体总结 30

1.4 本章总结 31

第2章 数据存储——内存 32

2.1 数据类型的存储 32

2.1.1 知识点题型 33

201 char *a[3][4]占
内存大小 33202 int *p[M]与
int (*p)[M] 33203 数组元素指针与数组
元素大小的计算 33204 #define 与 typedef
区别 34

205 sizeof(p) 35

206 关键字“static”修饰
变量与函数 36207 char 不等于
signed char 36208 unsigned 类型值与 0
的关系 37

209 不同数据类型的赋值 38

2.1.2 综合题型 39

210 验证 C 语言内存
分配 39

2.2 内存相关的 bug 41

2.2.1 知识点题型 41

211 返回栈地址 41

212 未初始化与内存访
问越界 41

213	使用已释放的内存	42	第 4 章 程序的基本单位——函数	97
214	数组越界	43	4.1 自定义函数	97
215	内存泄漏之内存 释放顺序	43	4.1.1 知识点题型	97
216	内存泄漏之重新赋值	44	401 函数声明与定义	97
217	内存泄漏之谨慎使用 函数返回值	45	402 十六进制转十进制	99
218	内存泄漏之退出 函数路径	47	403 函数纠错	101
2.2.2 综合题型		48	404 找出子串的个数	102
219	时间 GeneralizedTime 编码格式	48	405 指针函数与 函数指针	103
220	浮点数的存储格式	49	406 回调函数	104
221	实现小应用—— 万年历	51	407 可变参数函数	106
2.3 本章总结		54	408 函数调用	108
第 3 章 程序的思想——算法		56	409 内联 (inline) 函数	110
3.1 基本数据结构		56	410 main()函数的命令行 参数	112
3.1.1 知识点题型		56	4.1.2 综合题型	114
301 栈-迷宫问题		56	411 hexdump()函数	114
302 队列——旅行家的 预算		62	412 Sequence of Function Execution	115
303 链表——约瑟夫问题		69	4.2 库函数	116
304 二叉树遍历		71	4.2.1 知识点题型	117
305 Modbus CRC		72	413 调试相关的宏	117
3.1.2 综合题型		74	414 大小写字母的转换	117
306 汉诺塔问题		74	415 文件操作	119
307 找符合规则的数		77	416 数学库	122
3.2 查找与排序		78	4.2.2 综合题型	123
3.2.1 查找		78	417 All Combinations of a Word	123
308 二分法查找		78	4.2.3 库函数总结	128
309 Hash 查找		80	4.3 本章总结	128
3.2.2 排序		83	第 5 章 数据的传递——网络	130
310 选择排序		83	5.1 网络基础	130
311 冒泡排序		84	5.1.1 知识点题型	130
312 插入排序		86	501 网络字节序	130
313 快速排序		87	502 IP 地址转换	131
3.3 综合试题		90	503 设计数据报文头 结构	133
314 Colored Cubes		90	504 IP 头校验和	135
3.4 本章总结		95	5.1.2 综合题型	137
			505 RC4 Algorithm	137

506 解析 URL	141	615 指针偏移计算	193
5.2 Linux Socket 编程	143	616 判断链表是否 存在环	194
5.2.1 TCP Socket APIs	143	617 const 与指针	196
5.2.2 UDP Socket APIs	146	618 指向绝对地址执行	196
5.2.3 典型试题	148	619 同名变量（全局变量 与局部变量）	197
507 实现简单 ping 命令	148	620 gets()与 fgets()	198
508 实现 HTTP GET	156	621 头文件重复引用	199
509 Blocking 和 Non- blocking	160	6.2 可移植性	200
510 select()函数用法	161	622 u8、u16、u32 和 u64	200
511 fork()与 pthread_ create()	166	623 条件编译	200
5.3 综合试题	170	624 转义字符问题	202
512 SYN 洪流攻击	170	625 提高可移植性 常见技巧	203
5.4 本章总结	179	6.3 代码优化	203
第 6 章 专用的计算机系统—— 嵌入式	181	626 i--与 i < 10	203
6.1 嵌入式编程基础	181	627 switch 代替 if··else··	204
601 对寄存器的位操作	181	628 代码优化的 常见技巧	205
602 register 关键字	182	6.4 综合试题	205
603 *(volatile unsigned long *)0xfff40080;	183	629 模拟 Modbus 协议	205
604 #ifdef、#if、#if defined 和 #ifndef	183	6.5 本章总结	209
605 -1 在内存中的表示	184	第 7 章 终结语	211
606 找出重复的数	185	7.1 编程习惯	211
607 #define 中的 “##” 与 “#”	186	7.2 Bug 案例	214
608 计算一个整数中含 1 的比特数	187	7.3 学习技巧	217
609 递归计算	188	7.4 本章总结	220
610 数值溢出运算	189	附录 A ASCII 码表	221
611 定义 boolean 类型	189	附录 B C 语言运算符优先级	223
612 for()、while() 与 do{ }while()	189	附录 C 标准 C 头文件	225
613 位域应用	190	参考文献	226
614 break 与 continue 的 区别	191		

第0章 笔试题自测

引言：在学习本书之前，笔者建议读者先感受一下企业招聘时的部分真实 C 笔试、面试题，这不仅仅是一个自我测验，更是下一步学习的目标、导航。

基础题

试题 1-1: Please select the correct result:

```
printf("%d,%d",f1(),f2());
```

What is the sequence of function execution of the above statement?

A. printf, f1(), f2()

B. printf, f2(), f1()

C. f1(), f2(), printf

D. f2(), f1(), printf

试题 1-2: 请指出该函数存在哪些问题？

```
1 char *func()
2 {
3     char *a;
4     a = (char*)malloc(5);
5     strcpy(a, "ABCDE");
6     a = "CDE";
7     return a;
8 }
```

试题 1-3: 请问 `int *p[M]` 与 `int (*p)[M]` 的区别是什么？

试题 1-4: 以下代码打印的结果是（假设运行在 32 位系统上）（ ）。

```
1 #include <stdio.h>
2 int main()
3 {
4     struct st_t
5     {
6         int    status;
7         short* pdata;
8         char   errstr[32];
9     };
10    struct st_t st[16];
11    char* p = (char*)(st[2].errstr + 32);
12    printf("%d", (p - (char*)(st)));
```

```
13     return 0;
14 }
```

A. 32 B. 114 C. 120 D. 1112

试题 1-5: 请问下述代码是否存在问题? 为什么?

```
1  #ifndef __MSDOS__
2  const char name[] = "/root/new/table";
3  #else
4  const char name[] = "\\root\\new\\table";
5  #endif
```

试题 1-6: What is the type of the variable b in the following declaration?

```
#define FLOATPTR    float*
FLOATPTR a, b;
```

A. float
B. float pointer
C. int
D. int pointer

试题 1-7: Given char (*(x())[5])(); please describe the meaning of x.

试题 1-8: 分析下属程序的结果输出, 并给予解释。

```
1  #include <stdio.h>
2
3  struct DATA
4  {
5      long num;
6      char *name;
7      short int data;
8      char ha;
9      short ba[5];
10 } *p;
11
12 int main()
13 {
14     p = (struct DATA *)0x10000000;
15     printf("%x\n", p + sizeof(*p));
16     printf("%x\n", p + 1);
```

```

17     printf("%x\n", (unsigned long)p + 1);
18     printf("%x\n", (int *)p + 1);
19     printf("%x\n", (char *)p + 1);
20
21     return 0;
22 }

```

试题 1-9: 请问执行如下一段代码后, sum 的值为多少?

```

char chr = 127;
int sum = 100;
chr += 1;
sum += chr;

```

试题 1-10: 请分析下述代码潜在的问题。

```

1     free(hotbmp->pHotBitmapCacheMemory);
2     free(hotbmp);
3     hotbmp->pHotBitmapCacheMemory = NULL;
4     hotbmp->HotBitmapCacheLength = 0;
5     hotbmp = NULL;

```

综合题

试题 2-1: 请编写一个函数, 完成在一个字符串中查找子字符串出现的次数。例如: 在字符串 “President Obama has announced his support for India's bid for a permanent place on the United Nations Security Council. Obama was addressing the Indian parliament.” 中查找子串 “Obama” 出现的次数。

试题 2-2: You have four colored cubes. Each side of each cube is a single color, and there are four colors: blue (B), red (R), green (G) and yellow (Y). Describing the six faces as front, back, left, right, top, bottom, the cube colors are:

Cube	Front	Back	Left	Right	Top	Bottom
1	R	B	G	Y	B	Y
2	R	G	G	Y	B	B
3	Y	B	R	G	Y	R
4	Y	G	B	R	R	R

The objective is to find ways to stack the four cubes as a vertical column so that each side

of the column is showing all four colors. Write a C program to find all successful permutations.

试题 2-3: 一个旅行家想驾驶汽车以最少的费用从一个城市 A 到另一个城市 B (假设出发时油箱是空的)。给定两个城市之间的距离 $D1$ 、汽车油箱的容量 C (以升为单位)。每升汽油能行驶距离 $D2$ 、出发点的每升汽油价格 P 和沿途油站数 N (可以是 0)、油站 I 离出发点的距离 DI 、油站 I 每升汽油价格 PI ($I=1,2,3,\dots,N$)。编程找出一种加油方案, 使费用最少, 输出这个最少的费用值, 计算结果四舍五入至小数点后两位。如果无法到达目的地, 则输出“No Solution”。例如:

输入:

```
275.60, 11.90, 27.40, 2.80, 2 //各值分别代表 D1、C、D2、P、N
102.00, 2.90 //D1, P1
220.00, 2.20 //D2, P2
```

输出:

```
Cost=26.95 //该数据表示最小费用
```

第1章 灵魂——指针

引言：C 语言编程的精髓就是对指针的精巧使用，利用指针变量可以表示各种类型的数据结构，并能像汇编语言一样处理内存地址，可以编出精练而高效的程序。指针极大地丰富了C语言的功能，把指针比作C语言的灵魂一点都不为过。要想成为一名优秀的C程序设计人员，对指针必须要有一个深入而全面的理解。既然指针如此重要，那么各大企业在面试中对该技术的考查几乎是必须的。让我们就从C编程中最有传奇色彩的指针开始学习吧！

本章学习概要

1.1 指针与字符串

1.1.1 知识点题型

1.1.2 综合题型

1.2 指针与数组

1.2.1 知识点题型

1.2.2 综合题型

1.2.3 数组总结

1.3 指针与结构体

1.3.1 知识点题型

1.3.2 综合题型

1.3.3 结构体总结

1.1 指针与字符串

字符串是C语言编程中最常见的数据形式之一，它是以'\0'字符或NULL结尾的一维字符数组。字符串是对具体数据以字符形式的描述，在内存中以一维数组的形式存储。其实在日常生活中，很多信息都是以字符形式保存的，比如联系人信息、短消息、账号、网址等。

1.1.1 知识点题型

本书中“知识点题型”子章节主要以C语言的专业知识点为分析和讲述对象，通常是以频繁被使用的典型技术为切入点，笔者力图以点代面、抛砖引玉，最终达到触类旁通的目的。

101 p++

➤ 描述

请分析下述代码的输出结果。

```
1 int main()  
2 {  
3     char *p1 = "China";
```



```

4     char *p2, *p3;
5
6     p2 = (char*)malloc(20);
7     memset(p2, 0, 20);
8     while(*p2++ = *p1++);
9     printf("%s\n", p2);
10
11     return 0;
12 }

```

► 解析

输出为空。上述代码的本意是将字符串“China”复制到 p2 指针指向的内存区域，然后打印 p2 指向的内存区域中的字符串。但实际的运行结果是什么也没有打印。因为第 8 行代码，p1 与 p2 同时自增，所以当复制完 p1 中的字符串后，p2 也指向了“China”字符串的下一个地址处，故内容为空（在 ASCII^[1]码中 0 代表'\0'），如图 1-1 所示。

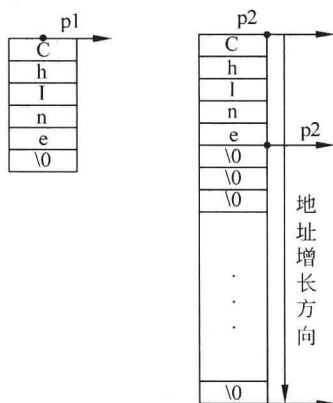


图 1-1 p1 与 p2 指针中字符串的存储情况

P 指针知识点扩充见表 1.1。

表 1.1 指针知识点扩充

++(*p)	先取 p 指针指向的内容，然后给内容加 1，加 1 的操作在当前生效。该语句与++*p 等价
(*p)++	先取 p 指针指向的内容，然后给内容加 1，加 1 的操作在该语句执行后生效。该语句与*p++ 等价
*(p++)	p 指针加 1，加 1 的操作在该语句执行后生效，然后取当前 p 指向的内容。该语句与*p++ 等价
(++p)	p 指针加 1，加 1 的操作在当前生效，然后取当前 p 加 1 后指向的内容。该语句与++p 等价

[1] ASCII (American Standard Code for Information Interchange, 美国信息互换标准代码) 是基于拉丁字母的一套电脑编码系统。它主要用于显示现代英语和其他西欧语言。它是现今最通用的单字节编码系统，并等同于国际标准 ISO/IEC 646。

102 打印内存地址

➤ 描述

请问在 `printf` 打印函数中一般用什么形式打印变量内存地址？

➤ 解析

该题目几乎是 C 语言编程中必须使用的一项基本知识。最常用的打印地址方式是 `%p` 和 `0x%x` 两种。如下面的代码用来打印局部变量 `a` 的内存地址。

```
#include <stdio.h>
int main()
{
    int a;
    printf("%p\n",&a);
    printf("0x%x\n",&a);
    return 0;
}
```

上述代码的输出结果是：

```
0xbffff664
0xbffff664
```

`%p` 与 `%x` 的区别是：`%p` 打印出的地址自动在地址前面加 `0x` 前缀，表示以十六进制打印；而 `%x` 仅仅是以十六进制打印，前面不会自动加 `0x`。所以使用 `%x` 打印地址或其他数值时，通常使用 `0x%x` 的形式打印。

103 `sprintf()`与 `snprintf()`

➤ 描述

请问 `printf()`、`sprintf()`与 `snprintf()`函数有什么区别？

➤ 解析

C 语言中的 `printf()`函数是每位学习 C 语言最熟悉不过的函数之一，其功能是按要求打印输出。一些读者可能会对 `sprintf()`或 `snprintf()`感到有些陌生。其实 `sprintf()`在实际编程中应用非常广泛。`printf()`是将一个格式化的字符串输出到屏幕；而 `sprintf()`是将一个格式化的字符串输出到一个指定的数组中，`snprintf()`是 `sprintf()`的限制字符数量的一种特殊用法。这 3 个函数的标准库函数原型如下：

```
int printf(const char *format, ...);
int sprintf(char *str, const char *format, ...);
int snprintf(char *str, size_t size, const char *format, ...);
```

对于 `printf` 这种 IT 界人士都知道的函数笔者就不在此赘述。我们通过下面的代码例子来学习 `snprintf()`和 `sprintf()`的基本用法。

```
#include <stdio.h>
int main()
```