



全国计算机等级考试

四合一过关训练

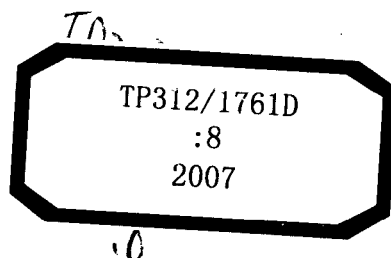
—— 二级C语言程序设计

笔试+上机+模拟题+真题

全国计算机等级考试命题研究组 编

南开大学出版社

全国计算机等级考试



四合一过关训练

二级 C 语言程序设计

全国计算机等级考试命题研究组 编

南开大学出版社

天 津

图书在版编目(CIP)数据

全国计算机等级考试四合一过关训练. 二级 C 语言程序设计 / 全国计算机等级考试命题研究组编. —天津:南开大学出版社, 2007. 11

ISBN 978-7-310-02770-5

I. 全… II. 全… III. ①电子计算机—水平考试—习题
②C 语言—程序设计—水平考试—习题 IV. TP3-44

中国版本图书馆 CIP 数据核字(2007)第 155648 号

版权所有 侵权必究

南开大学出版社出版发行

出版人: 肖占鹏

地址: 天津市南开区卫津路 94 号 邮政编码: 300071

营销部电话: (022)23508339 23500755

营销部传真: (022)23508542 邮购部电话: (022)23502200

*

河北昌黎太阳红彩色印刷有限责任公司印刷

全国各地新华书店经销

*

2007 年 11 月第 1 版 2007 年 11 月第 1 次印刷

787×1092 毫米 16 开本 16.875 印张 424 千字

定价: 30.00 元

如遇图书印装质量问题, 请与本社营销部联系调换, 电话: (022)23507125

内容提要

本书提供了全国计算机等级考试二级 C 语言程序设计的笔试和上机模拟试卷及真题,并给出精准的答案、详细的分析、考核的知识点、重点难点。主要内容有:笔试全真模拟试卷及解析;笔试真题及解析;上机全真模拟试题及解析;上机真题及解析;备考策略。

本书配套光盘主要内容有:(1)上机考试的全真模拟环境,可在此环境中练习 100 套上机题,进行答题和评分,以此进行考前强化训练。(2)笔试考试的全真模拟环境,可在此练习大量笔试题。(3)上机考试过程的录像动画演示,从登录、答题到交卷,均有指导教师的全程语音讲解;(4)本书上机试题的源文件。

本书针对参加全国计算机等级考试二级 C 语言程序设计的考生,同时也可作为普通高校、大专院校、成人高等教育以及相关培训班的练习题和考试题使用。

全国计算机等级考试专业网站百分网 <http://www.baifen100.com> 为读者提供全方位的技术支持。

前言

全国计算机等级考试（National Computer Rank Examination, NCRE）是由教育部考试中心主办，用于考查应试人员的计算机应用知识与能力的考试。本考试的证书已经成为许多单位招聘员工的一个必要条件，具有相当的“含金量”。

为了帮助考生更顺利地通过计算机等级考试，我们做了大量市场调查，根据考生的备考体会，以及培训教师的授课经验，推出了《四合一过关训练——二级 C 语言程序设计》。

本书主要特点

本书主要特点如下：

- **选题经典，解析详尽。**书中所选题目是极具代表性的经典试题，形式和难度都与真题类似，并涵盖了方方面面的考点。透彻深入的详尽解析可使您触类旁通，掌握解答相关问题的关键。
- **海量试题，物超所值。**书中提供了几十套模拟题和最新真题；光盘中还有 100 套（300 道题）历届上机真题和大量笔试题，可检验知识的掌握程度和训练答题的速度和准确性，以练促学，做到心中有数。
- **模拟考场，真实感受。**光盘中的上机全真模拟系统与真实考试环境相同，却比真实考试多了自动阅卷、自动评分和详尽解析的功能。您在这里可以感受真实的考试氛围，做到胸有成竹。
- **备考策略，简明实用。**每年，我们都收到一些考生的反馈信息，比如，考生的源代码写对了，上机考试却得 0 分，原因是什么呢？为此，我们在附录中为您准备了备考策略，使您能够避免发生类似的问题。这里还提供了答题技巧、注意事项等考试必备知识。
- **视频引导，直观详细。**附赠光盘包含上机操作过程的多媒体教学演示，其流畅的画质、简便的控制按钮、详实的步骤提示，可使您在不经意间迅速掌握要领。

本书主要内容

对于备战等级考试而言，做题，是进行考前冲刺的最佳方式。通过实际练习，可检验自己是否真正掌握了相关知识点，了解考试重点，并且根据需要再对知识结构的薄弱环节进行强化。本书的第一部分到第四部分分别是笔试全真模拟试卷及解析、笔试真题及解析、上机全真模拟试题及解析以及上机真题及解析。附录中的备考策略，说明了选择题和填空题的答题技巧、上机考试注意事项、上机考试过程等考试必备知识。

本书配套光盘主要内容有：

- （1）上机考试的全真模拟环境，可在此环境中练习 100 套上机题，进行答题和评分，以此进行考前强化训练。
- （2）笔试考试的全真模拟环境，可在此练习大量考题，并查看评分。

(3) 上机考试过程的录像动画演示, 从登录、答题到交卷, 均有指导教师的全程语音讲解。

(4) 本书上机试题的源文件。

本书针对参加全国计算机等级考试二级 C 语言程序设计的考生, 同时也可作为普通高校、大专院校、成人高等教育以及相关培训班的练习题和考试题使用。

为了保证本书及时面市和内容准确, 很多朋友做出了贡献, 陈河南、贺民、许伟、侯佳宜、贺军、于樊鹏、戴文雅、戴军、李志云、陈安南、李晓春、王春桥、王雷、韦笑、龚亚萍、冯哲、邓卫、唐玮、魏宇、李强等老师付出了很多辛苦, 在此一并表示感谢!

在学习的过程中, 您如有问题或宝贵意见和建议, 请通过电子邮件与我们联系。或登录百分网, 在“书友论坛”与我们共同探讨。

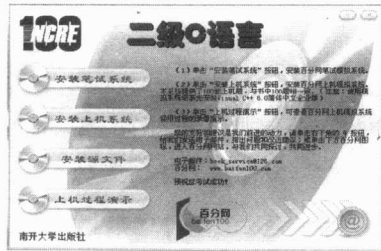
电子邮件: book_service@126.com

百分网: www.baifen100.com

全国计算机等级考试命题研究组

2007 年 10 月

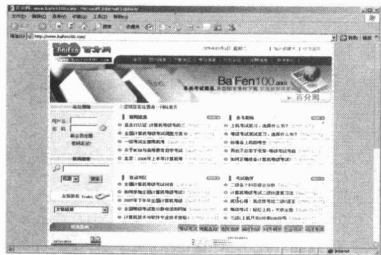
配套光盘说明



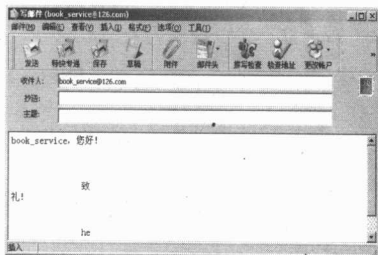
光盘初始启动界面, 可选择安装笔试题系统、上机系统、查看上机操作过程, 安装源文件



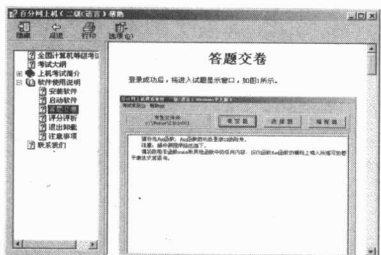
上机操作过程的录像演示, 有指导教师的全程语音讲解



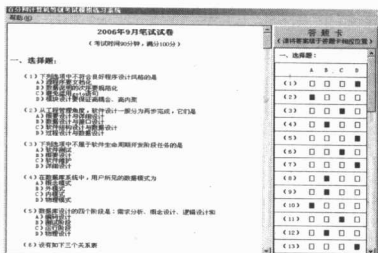
单击光盘初始界面的  图标, 可进入百分网, 您可以在此与我们共同探讨问题



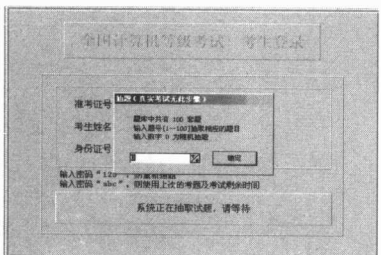
单击光盘初始界面左下角的  图标, 您可以给我们发送邮件, 提出您的建议和意见



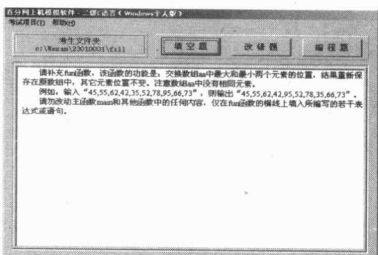
从“开始”菜单可启动帮助系统, 在这里可看到考试简介、考试大纲以及详细的软件使用说明



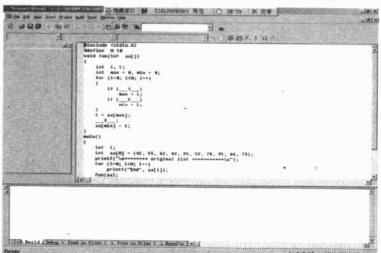
笔试题系统中, 您可以练习大量笔试题, 并查看评分结果



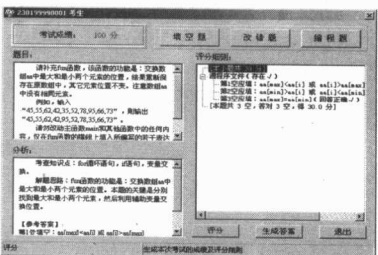
您可以随机抽题, 也可以指定固定的题目



浏览题目界面, 查看考试题目, 单击“考试项目”开始答题



在实际环境答题, 完成后单击工具栏中的“交卷”按钮



答案和分析界面, 查看所考核题目的答案和分析

目 录

第一部分 笔试全真模拟试卷及解析	1
第1套全真模拟试卷	1
第1套全真模拟试卷解析	11
第2套全真模拟试卷	17
第2套全真模拟试卷解析	27
第3套全真模拟试卷	34
第3套全真模拟试卷解析	44
第4套全真模拟试卷	51
第4套全真模拟试卷解析	60
第5套全真模拟试卷	66
第5套全真模拟试卷解析	77
第6套全真模拟试卷	83
第6套全真模拟试卷解析	93
第7套全真模拟试卷	99
第7套全真模拟试卷解析	109
第8套全真模拟试卷	116
第8套全真模拟试卷解析	125
第二部分 笔试真题及解析	133
2006年4月笔试真题	133
2006年4月笔试真题解析	142
2006年9月笔试真题	150
2006年9月笔试真题解析	161
2007年4月笔试真题	168
2007年4月笔试真题解析	178
2007年9月笔试真题	189
2007年9月笔试真题解析	199
第三部分 上机全真模拟题及解析	207
第1套上机全真模拟题	207
第1套上机全真模拟题解析	209
第2套上机全真模拟题	211
第2套上机全真模拟题解析	213
第3套上机全真模拟题	215
第3套上机全真模拟题解析	216
第4套上机全真模拟题	218

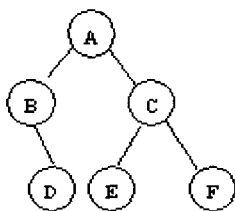
第 4 套上机全真模拟题解析	220
第 5 套上机全真模拟题	221
第 5 套上机全真模拟题解析	224
第 6 套上机全真模拟题	226
第 6 套上机全真模拟题解析	228
第 7 套上机全真模拟题	229
第 7 套上机全真模拟题解析	231
第 8 套上机全真模拟题	232
第 8 套上机全真模拟题解析	234
第四部分 上机真题及解析	237
第 1 套上机真题	237
第 1 套上机真题解析	239
第 2 套上机真题	241
第 2 套上机真题解析	243
第 3 套上机真题	245
第 3 套上机真题解析	247
第 4 套上机真题	249
第 4 套上机真题解析	251
附录 应试策略	253
考生须知	253
笔试答题技巧	253
笔试答题注意事项	253
选择题答题技巧	254
填空题答题技巧	254
上机考试指南	254
复习经验	254
上机考试过程	256

第一部分 笔试全真模拟试卷及解析

第 1 套全真模拟试卷

一、选择题

- (1) 下列选项中不属于算法的特性的是
A) 确定性 B) 可行性 C) 有输出 D) 无穷性
- (2) 下列叙述中正确的是
A) 线性表是线性结构 B) 栈与队列是非线性结构
C) 线性链表是非线性结构 D) 二叉树是线性结构
- (3) 设有下列二叉树:



- 对此二叉树中序遍历的结果为
A) ABCDEF B) DAECF C) BDAECF D) DBEFC A
- (4) 结构化程序设计主要强调的是
A) 程序的规模 B) 程序的易读性
C) 程序的执行效率 D) 程序的可移植性
- (5) 在面向对象设计中, 对象有很多基本特点, 其中“从外面看只能看到对象的外部特性, 而对象的内部对外是不可见的”这一性质指的是对象的
A) 分类性 B) 标识惟一性 C) 多态性 D) 封装性
- (6) 用黑盒技术测试用例的方法之一为
A) 因果图 B) 逻辑覆盖 C) 循环覆盖 D) 基本路径测试
- (7) 在进行单元测试时, 常用的方法是
A) 采用白盒测试, 辅之以黑盒测试 B) 采用黑盒测试, 辅之以白盒测试
C) 只使用白盒测试 D) 只使用黑盒测试
- (8) 下面四个选项中, 不属于数据库管理系统提供的数据库语言的是
A) 数据定义语言 B) 数据查询语言
C) 数据操纵语言 D) 数据控制语言
- (9) 在关系数据库模型中, 通常可以把 () 称为属性, 其值称为属性值。
A) 记录 B) 基本表 C) 模式 D) 字段
- (10) 实体联系模型中实体与实体之间的联系不可能是
A) 一对一 B) 多对多 C) 一对多 D) 一对零
- (11) 用 C 语言编写的代码程序
A) 可立即执行 B) 是一个源程序

- C) 经过编译即可执行 D) 经过编译解释即可执行
- (12) 以下叙述中正确的是
- A) C 语言的源程序不必通过编译就可以直接运行
 B) C 语言中的每条可执行语句最终都将被转换成二进制的机器指令
 C) C 源程序经编译形成的二进制代码可以直接运行
 D) C 语言中的函数不可以单独进行编译
- (13) 以下定义语句中正确的是
- A) char a='A' b='B'; B) float a=b=10.0;
 C) int a=10,*b=&a; D) float *a,b=&a;
- (14) 有以下程序:
- ```
main()
{ int a; char c=10;
 float f=100.0; double x;
 a=f/=c*=(x=6.5);
 printf("%d %d %3.1f %3.1f\n",a, c, f, x);
}
```
- 程序运行后的输出结果是
- A) 1 65 1 6.5                      B) 1 65 1.5 6.5  
 C) 1 65 1.0 6.5                      D) 2 65 1.5 6.5
- (15) 数据库设计的根本目标是要解决
- A) 数据共享问题                      B) 数据安全问题  
 C) 大量数据存储空间                      D) 简化数据维护
- (16) 设有定义: float a=2,b=4,h=3;, 以下 C 语言表达式中与代数式  $\frac{1}{2}(a+b)h$  计算结果不相符的是
- A) (a+b)\*h/2                      B) (1/2)\*(a+b)\*h                      C) (a+b)\*h\*1/2                      D) h/2\*(a+b)
- (17) 有以下程序
- ```
main()
{
  int m=0256,n=256;
  printf("%o %o\n",m,n);
}
```
- 程序运行后的输出结果是
- A) 0256 0400 B) 0256 256 C) 256 400 D) 400 400
- (18) 数据库系统的核心是
- A) 数据模型 B) 数据库管理系统 C) 数据库 D) 数据库管理员
- (19) 以下叙述中错误的是
- A) 用户所定义的标识符允许使用关键字
 B) 用户所定义的标识符应尽量做到“见名知意”
 C) 用户所定义的标识符必须以字母或下划线开头
 D) 用户定义的标识符中, 大、小写字母代表不同标识
- (20) 有以下程序:
- ```
main()
{
 char k; int i;
 for(i=1;i<3;i++)
 {
 scanf("%c",&k);
 }
}
```

```

switch(k)
{
case '0': printf("another\n");
case '1': printf("number\n");
}
}

```

程序运行时,从键盘输入:01<回车>,程序执行后的输出结果是

- |            |            |            |           |
|------------|------------|------------|-----------|
| A) another | B) another | C) another | D) number |
| number     | number     | number     | number    |
|            | another    | number     |           |

(21) 以下叙述中正确的是

- A) 用 C 程序实现的算法必须要有输入和输出操作  
 B) 用 C 程序实现的算法可以没有输出但必须要有输入  
 C) 用 C 程序实现的算法可以没有输入但必须要有输出  
 D) 用 C 程序实现的算法可以既没有输入也没有输出

(22) 有以下程序

```

main()
{ int i=0,s=0;
 do{
 if(i%2){i++;continue;}
 i++;
 s+=i;
 } while(i<7);
 printf("%d\n",s);
}

```

执行后输出的结果是

- A) 16                      B) 12                      C) 28                      D) 21

(23) 已定义 ch 为字符型变量,以下赋值语句中错误的是

- A) ch='\';                  B) ch=62+3;                  C) ch=NULL;                  D) ch='\xaa';

(24) 有以下程序

```

main()
{ char a,b,c,d;
 scanf("%c,%c,%d,%d",&a,&b,&c,&d);
 printf("%c,%c,%c,%c\n",a,b,c,d);
}

```

若运行时从键盘上输入:6,5,65,66<回车>。则输出结果是

- A) 6,5,A,B                  B) 6,5,65,66                  C) 6,5,6,5                  D) 6,5,6,6

(25) 以下能正确定义二维数组的是

- A) int a[][3];                      B) int a[][3]={2\*3};  
 C) int a[][3]={};                      D) int a[2][3]={{1},{2},{3,4}};

(26) 以下程序的输出结果是

```

#include <stdio.h>
main()
{ int i=0,a=0;
 while(i<20)
 { for(;;)
 { if((i%10)==0) break;
 else i--;
 }
 i+=11; a+=i;
 }
}

```

```
printf("%d\n",a);
}
```

- A) 21                      B) 32                      C) 33                      D) 11

(27) 若程序中已包含头文件 `stdio.h`, 以下选项中, 正确运用指针变量的程序段是

- A) `int *i = NULL;`  
`scanf("%d", i);`  
 B) `float *f = NULL;`  
`*f = 10.5;`  
 C) `char t = 'm', *c = &t;`  
`*c = &t;`  
 D) `long *L;`  
`L = '\0';`

(28) 有以下程序:

```
#include <stdio.h>
main()
{ printf("%d\n", NULL); }
```

程序运行后的输出结果是

- A) 0                      B) 1                      C) -1                      D) NULL 没定义, 出错

(29) 有以下程序

```
main()
{ char *s[]={"one","two","three"},*p;
 p=s[1];
 printf("%c,%s\n",*(p+1),s[0]);
}
```

执行后输出结果是

- A) n,two                      B) t,one                      C) w,one                      D) o,two

(30) 以下不能正确定义二维数组的选项是

- A) `int a[2][2] = {{1}, {2}};`  
 B) `int a[ ][2] = {1, 2, 3, 4};`  
 C) `int a[2][2] = {{1}, 2, 3};`  
 D) `int a[2][ ] = {{1, 2}, {3, 4}};`

(31) 设有定义: `int a=2,b=3,c=4;`, 则以下选项中值为 0 的表达式是

- A) `(!a==1)&&(!b==0)`  
 B) `(a<b)&&!c||1`  
 C) `a && b`  
 D) `all(b+b)&&(c-a)`

(32) 设有以下定义和语句

```
int a[3][2]={1,2,3,4,5,6}, *p[3];
p[0]=a[1];
```

则 `*(p[0]+1)` 所代表的数组元素是

- A) `a[0][1]`                      B) `a[1][0]`                      C) `a[1][1]`                      D) `a[1][2]`

(33) 有以下程序

```
main()
{
 int aa[4][4]={{1,2,3,4},{5,6,7,8},{3,9,10,2},{4,2,9,6}};
 int i,s=0;
 for(i=0;i<4;i++) s+=aa[i][1];
 printf("%d\n",s);
}
```

程序运行后的输出结果是

- A) 11                      B) 19                      C) 13                      D) 20

(34) 有以下程序

```
int f1(int x,int y){ return x>y?x:y; }
int f2(int x,int y){ return x>y?y:x; }
main()
{
 int a=4,b=3,c=5,d=2,e,f,g;
 e = f2(f1(a,b), f1(c,d)); f=f1(f2(a,b), f2(c,d));
 g = a+b+c+d-e-f;
```

```
printf("%d,%d,%d\n",e,f,g);
}
```

程序运行后的输出结果是

- A) 4,3,7      B) 3,4,7      C) 5,2,7      D) 2,5,7

(35) 设变量已正确定义, 则以下能正确计算  $f=n!$  的程序段是

- A)  $f=0;$       B)  $f=1;$   
     for( $i=1;i \leq n;i++$ )  $f*=i;$       for( $i=1;i < n;i++$ )  $f*=i;$   
 C)  $f=1;$       D)  $f=1;$   
     for( $i=n;i > 1;i--$ )  $f*=i;$       for( $i=n;i \geq 2;i--$ )  $f*=i;$

(36) 以下叙述中正确的是

- A) 全局变量的作用域一定比局部变量的作用域范围大  
 B) 静态 (static) 类别变量的生存期贯穿于整个程序的运行期间  
 C) 函数的形参都属于全局变量  
 D) 未在定义语句中赋初值的 auto 变量和 static 变量的初值都是随机值

(37) 设有以下语句:

```
typedef struct S
{ int g; char h; } T;
```

则下面叙述中正确的是

- A) 可用 S 定义结构体变量      B) 可以用 T 定义结构体变量  
 C) S 是 struct 类型的变量      D) T 是 struct s 类型的变量

(38) 以下叙述中错误的是

- A) 对于 double 类型数组, 不可以直接用数组名对数组进行整体输入或输出  
 B) 数组名代表的是数组所占存储区的首地址, 其值不可改变  
 C) 当程序执行中, 数组元素的下标超出所定义的下标范围时, 系统将给出“下标越界”的出错信息  
 D) 可以通过赋初值的方式确定数组元素的个数

(39) 有以下程序

```
int fa(int x)
{ return x*x; }
int fb(int x)
{ return x*x*x; }
int f(int (*f1)(), int (*f2)(), int x)
{ return f2(x)-f1(x); }
main()
{ int i;
 i=f(fa,fb,2); printf("%d \n",i);
}
```

程序运行后的输出结果是

- A) -4      B) 1      C) 4      D) 8

(40) 有以下程序:

```
struct s
{ int x,y; } data[2]={10,100,20,200};
main()
{ struct s *p=data;
 printf("%d\n",++(p->x));
}
```

程序运行后的输出结果是

- A) 10      B) 11      C) 20      D) 21

(41) 有以下程序

```
void ss(char *s,char t)
```

```

{
 while(*s)
 {
 if(*s==t) *s=t-'a'+'A';
 s++;
 }
}
main()
{
 char str1[100]="abcddfefdbd",c='d';
 ss(str1,c); printf("%s\n",str1);
}

```

程序运行后的输出结果是

A) ABCDDEFEDBD

B) abcDDfefDbD

C) abcAAfefAbA

D) Abcddfefdbd

(42) 有以下程序

```

main(int argc, char *argv[])
{
 int n,i=0;
 while(argv[1][i]!='\0')
 {
 n=fun(); i++;
 printf("%d\n",n*argc);
 }
}
int fun()
{
 static int s=0;
 s+=1;
 return s;
}

```

假设程序经编译、连接后生成可执行文件 exam.exe, 若键入以下命令  
exam 123 <回车>

则运行结果为

A) 6

B) 8

C) 3

D) 4

(43) 已定义以下函数:

```

fun (char* p2, char* p1)
{ while ((*p2=*p1) != '\0') {p1++;p2++;} }

```

函数的功能是

A) 将 p1 所指字符串复制到 p2 所指内存空间

B) 将 p1 所指字符串的地址赋给指针 p2

C) 对 p1 和 p2 两个指针所指字符串进行比较

D) 检查 p1 和 p2 两个指针所指字符串中是否有 '\0'

(44) 有以下程序

```

main()
{
 int x[] = {1, 3, 5, 7, 2, 4, 6, 0}, i, j, k;
 for (i = 0; i < 3; i++)
 for(j = 2; j >= i; j--)
 if (x[j+1] > x[j])
 {
 k = x[j];
 x[j] = x[j+1];
 x[j+1] = k;
 }
 for (i = 0; i < 3; i++)
 for (j = 4; j < 7-i; j++)
 if (x[j] > x[j+1])

```

```

 {
 k = x[j];
 x[j] = x[j+1];
 x[j+1] = k;
 }
 for (i = 0; i < 8; i++)
 printf("%d", x[i]);
 printf("\n");
}

```

程序运行后的输出结果是

- A) 75310246      B) 01234567      C) 76310462      D) 13570246

(45) 设有如下定义

```

struct ss
{
 char name[10];
 int age;
 char sex;
} std[3], * p=std;

```

下面各输入语句中错误的是

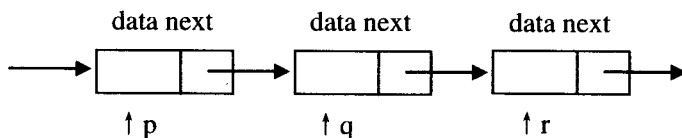
- A) scanf("%d",&(\*p).age);      B) scanf("%s",&std.name);  
 C) scanf("%c",&std[0].sex);      D) scanf("%c",&(p->sex));

(46) 有以下结构体说明和变量定义, 如图所示, 指针 p、q、r 分别指向一个链表中的三个连续结点。

```

struct node
{
 int data;
 struct node *next;
} *p, *q, *r;

```



现要将 q 和 r 所指结点的先后位置交换, 同时要保持链表的连续, 以下错误的程序段是

- A) r->next=q; q->next=r->next; p->next=r;  
 B) q->next=r->next; p->next=r; r->next=q;  
 C) p->next=r; q->next=r->next; r->next=q;  
 D) q->next=r->next; r->next=q; p->next=r;

(47) 以下程序中函数 f 的功能是将 n 个字符串按由大到小的顺序进行排序。

```

#include <string.h>
void f(char p[][10], int n)
{
 char t[20]; int i, j;
 for(i=0; i<n-1; i++)
 for(j=i+1; j<n; j++)
 if(strcmp(p[i], p[j])<0)
 { strcpy(t, p[i]); strcpy(p[i], p[j]); strcpy(p[j], t); }
}
main()
{
 char p[][10]={"abc", "aabdfg", "abbd", "dcdb", "cd"}; int i;
 f(p, 5); printf("%d\n", strlen(p[0]));
}

```



程序运行后的输出结果是

A) 6

B) 4

C) 5

D) 3

(48)~(50) 以下程序的功能是：建立一个带有头结点的单向链表，并将存储在数组中的字符依次转存到链表的各个结点中，请从与下划线处号码对应的一组选项中选择出正确的选项。

```
#include <stdlib.h>
struct node
{ char data; struct node *next; };
(48) CreatList(char *s)
{
 struct node *h,*p,*q;
 h = (struct node *)malloc(sizeof(struct node));
 p=q=h;
 while(*s!='\0')
 {
 p = (struct node *)malloc(sizeof(struct node));
 p->data = (49);
 q->next = p;
 q = (50);
 s++;
 }
 p->next='\0';
 return h;
}
main()
{
 char str[]="link list";
 struct node *head;
 head = CreatList(str);
 ...
}
```

(48) A) char \*

B) struct node

C) struct node \*

D) char

(49) A) \*s

B) s

C) \*s++

D) (\*s)++

(50) A) p->next

B) p

C) s

D) s->next

## 二、填空题

- 算法的复杂度主要包括空间复杂度和\_\_\_\_\_复杂度。
- 在面向对象方法中，类的实例称为\_\_\_\_\_。
- 对于软件测试，从是否需要执行被测软件的角度，可以分为静态测试和动态测试。代码检查属于\_\_\_\_\_测试。
- 实体联系模型是一种常用的高级概念数据模型，而\_\_\_\_\_是实体联系模型中的核心。
- 一个项目具有一个项目主管，一个项目主管可管理多个项目，则实体“项目主管”与实体“项目”的联系属于\_\_\_\_\_。
- 若有语句

```
int i=-19,j=i%4;
printf("%d\n",j);
```

则输出的结果是\_\_\_\_\_。

- 以下程序运行后的输出结果是\_\_\_\_\_。

```
#define S(x) 4*x*x+1
main()
{ int i=6, j=8;
 printf("%d\n",S(i+j));
```