



全国最畅销品牌优势升级!
全国1001所高校学子的明智选择

**2014年
考试专用**

全国计算机等级考试 历年真题必练(含关键考点点评) ——二级C语言程序设计

(第4版)

全国计算机
等级考试命题研究组 **编写**

QUANGUO JISUANJI DENGJI KAOSHI MINGTI YANJIJUZU

实战真题是考试过关的捷径
(考试必备方法之一)



赠 模拟考试
光盘一张



北京邮电大学出版社
www.buptpress.com

全新内容

本书是全国计算机等级考试历年真题的汇编，旨在帮助考生了解考试题型、难度和命题规律。本书收录了2014年全国计算机等级考试历年真题，包括一级、二级、三级和四级。本书可作为考生备考的参考资料，也可作为教师的教学参考。

2014 年全国计算机等级考试历年真题必练

(含关键考点点评)

——二级 C 语言程序设计(第4版)

全国计算机等级考试命题研究组 编写

福建师范大学
图书馆
藏书印

1170652



T1170652

北京邮电大学出版社

·北京·

全国计算机等级考试命题研究组

内 容 简 介

本书根据最新全国计算机等级考试最新考试大纲,由教育考试研究中心通过对历年等级考试真题研究分析而成。本书提供10套真题供考生使用,真题根据最新考试形式编排,让考生熟悉真实考试流程。每套真题附有答案解析和关键考点点评,方便考生快速重温重点难点,迅速提高应试能力!本书配有光盘,光盘中的配套软件完全模拟真题考试环境,便于考生实战演练。本书可供全国计算机等级考试二级C语言考生复习使用,特别适合考前冲刺使用,同时也非常适合相关等级考试培训班用作培训教材。

图书在版编目(CIP)数据

2014年全国计算机等级考试历年真题必练:含关键考点点评. 二级C语言程序设计/全国计算机等级考试命题研究组编写. --4版. --北京:北京邮电大学出版社,2014.1

ISBN 978-7-5635-3711-2

I. ①2… II. ①全… III. ①电子计算机—水平考试—习题集②C语言—程序设计—水平考试—习题集 IV. ①TP3-44

中国版本图书馆 CIP 数据核字(2013)第 226121 号

书 名: 2014 年全国计算机等级考试历年真题必练(含关键考点点评)——二级 C 语言程序设计(第 4 版)

作 者: 全国计算机等级考试命题研究组

责任编辑: 满志文 姚 顺

出版发行: 北京邮电大学出版社

社 址: 北京市海淀区西土城路 10 号(邮编: 100876)

发 行 部: 电话: 010-62282185 传真: 010-62283578

E-mail: publish@bupt.edu.cn

经 销: 各地新华书店

印 刷: 北京联兴华印刷厂

开 本: 787 mm×1 092 mm 1/16

印 张: 10.25

字 数: 465 千字

版 次: 2014 年 1 月第 4 版 2014 年 1 月第 1 次印刷

ISBN 978-7-5635-3711-2

定价: 27.00 元

• 如有印装质量问题,请与北京邮电大学出版社发行部联系 •

前言

全国计算机等级考试是全国范围内应试考生人数最多、规模最大、最具有影响力的权威性国家级计算机类水平考试,很多企事业单位都把获得全国计算机等级考试证书作为人事考核、人才招聘、职称晋升的评定条件之一。全国计算机等级考试是一种水平性考试,历年真题具有极强的规律性和重复性,通过研究我们发现一个惊人的事实:几乎每年都有2~3题是以前考过的真题,约有72%是雷同的考点,有变化的新考题仅仅有约9%!也就是说,只要把考过的真题都做会,就能轻松过关!

本书自第1版推出以来,凭借“举一反三的真题解析、独一无二的关键考点点评、揭示命题规律的真题链接”在广大考生中引起强烈震撼,有读者来信评价本书为短平快过关必读圣经!考生的需求是我们服务的目标,在上一版的基础上,我们吸收了众多读者与专家的建议,隆重推出第4版。本书在第3版的基础上进行了如下修订:

- 细致排错。对全书细致入微地进行了审查,决不放过任何细小的错误,确保内容的正确性,以便考生复习时畅通无阻。
- 与最新考试同步。本书添加了最新考试真题,并对每个考题进行了详尽的解析,有助于考生把握考试规律,及时了解最新考试动态。
- 深入研究命题动态。本书根据最新考试大纲,对所有考点进行了系统地分类,使得本书考点全面,删除与考试无关的考点,帮助考生节约复习时间。

本套产品由考卷和配套多媒体学习光盘组成,其中考卷部分包括:10套全真试题+试题详细解析+关键考点点评。配套多媒体光盘部分模拟真题考试环境,便于考生实战演练。

本书具有如下特色:

- (1) 真题套数多,附有答案解析。本书提供10套真考试题库供考生使用。
- (2) 根据最新考试形式编排,让考生熟悉真实考试流程。
- (3) 答案解析,详略得当:试卷不仅给出了参考答案,而且一一予以解题分析,突出重点、难点,详略得当,力求通过解析的学习,强化理解、记忆。
- (4) 每套试题解析最后附有关键考点点评。同类图书一般是“试卷+解析”的风格,我们根据培训老师的实际培训经验,在每套试卷解析最后加了“关键考点点评”,对本套试卷中的难点、重点进行剖析,使考生能达到举一反三的功效;对重点考点进行链接,使考生重温了相关知识,备考更有信心。
- (5) 按考试频率分类精选10套操作题。通过对操作题库的透彻分析,把这些真题分成若干类,按考试频率的高低从每类中精选最有代表性的真题,从而做到了以点代面、跳出题海,为考生考试过关指明了一条捷径。
- (6) 装帧独特,便于学习。每套试题按“试卷+解析+点评”装成一份,非常适合考生每份试题按“练、学、查”方式实战,而且充分考虑到培训班的特点,方便教学使用。
- (7) 书盘结合,题量超大。配套光盘中提供多套试题,全真模拟环境,便于考生实战演练,适应最新考试形式。
- (8) 作者实力强。作者团队从事等级考试近10年的辅导、培训、命题、阅卷及编写之经验,有较高的权威性,图书质量有保障。

本书由全国计算机等级考试命题研究组主编,参与编写与考试研究、光盘制作的人员有:何光明、王珊珊、周海霞、江梅、陈海燕、杜兰、薛英、屠强、张石磊、李为健、赵明明、吴远、刘英英、吴涛涛、赵梨花、陈智、赵传申、吴婷、刘家琪、李海、骆健、张居晓、唐瑞华。

本书可供全国计算机等级考试二级C语言考生复习使用,特别适合考前冲刺使用,同时也非常适合相关等级考试培训班用作培训教材。预祝各位考试成功,如遇到疑难问题,可通过以下方式与我们联系:bjbaba@263.net。微博地址: <http://weibo.com/2297589741>。(也请参与我们的微博活动吧!活动如下:①关注@北邮等考,成为北邮等考的粉丝。②转发微博“北邮出版的等考图书刚买到,相信能成功。全国计算机等级考试复习资料首选北邮出版的”,并说出你购买图书、参加考试的心情和故事,也可以是生活中的乐趣。我们将给优秀粉丝送礼,一直有效啊。)

目 录

2013年9月全国计算机等级考试二级C语言程序设计

(共16页)

试卷	1
试卷答案解析	10
选择题关键点点评	13
操作题关键点点评	16

2013年3月全国计算机等级考试二级C语言程序设计

(共13页)

试卷	1
试卷答案解析	8
选择题关键点点评	11
操作题关键点点评	13

2012年9月全国计算机等级考试二级C语言程序设计

(共15页)

试卷	1
试卷答案解析	10
选择题关键点点评	13
操作题关键点点评	15

2012年3月全国计算机等级考试二级C语言程序设计

(共14页)

试卷	1
试卷答案解析	9
选择题关键点点评	12
操作题关键点点评	14

2011年9月全国计算机等级考试二级C语言程序设计

(共17页)

试卷	1
试卷答案解析	11
选择题关键点点评	14
操作题关键点点评	16

2011年3月全国计算机等级考试二级C语言程序设计

(共15页)

试卷	1
试卷答案解析	10
选择题关键点点评	13
操作题关键点点评	15

2010年9月全国计算机等级考试二级C语言程序设计

(共15页)

试卷	1
试卷答案解析	10
选择题关键点点评	13
操作题关键点点评	15

2010年3月全国计算机等级考试二级C语言程序设计

(共18页)

试卷	1
试卷答案解析	11
选择题关键点点评	15
操作题关键点点评	17

2009年9月全国计算机等级考试二级C语言程序设计

(共16页)

试卷	1
试卷答案解析	10
选择题关键点点评	13
操作题关键点点评	15

2009年3月全国计算机等级考试二级C语言程序设计

(共16页)

试卷	1
试卷答案解析	10
选择题关键点点评	13
操作题关键点点评	15

说明:由于原来二级C语言考试真题的选择题只有35题,现根据新大纲要求,我们在以往考试真题中精选部分选择题,将原来的35题扩充至40题,以符合最新考试形式。

2013 年 9 月全国计算机等级考试二级 C 语言程序设计

试 卷

(考试时间 120 分钟, 满分 100 分)

一、选择题(每小题 1 分, 共 40 分)

- 下列各题 A)、B)、C)、D) 四个选项中, 只有一个选项是正确的。
- (1) 一个栈的初始状态为空。现将元素 1、2、3、4、5、A、B、C、D、E 依次入栈, 然后再依次出栈, 则元素出栈的顺序是_____。
- A) 12345ABCDE B) EDCBA54321
C) ABCDE12345 D) 54321EDCBA
- (2) 下列叙述中正确的是_____。
- A) 循环队列有对头和队尾两个指针, 因此, 循环队列是非线性结构
B) 在循环队列中, 只需要对头指针就能反应队列中元素的动态变化
C) 在循环队列中, 只需要对尾指针就能反应队列中元素的动态变化
D) 循环队列中元素的个数是由对头指针和队尾指针共同决定
- (3) 在长度为 n 的有序线性表中进行二分查找, 最坏情况下需要比较的次数是_____。
- A) $O(n)$ B) $O(n_2)$ C) $O(\log_2 n)$ D) $O(n \log_2 n)$
- (4) 下列叙述中正确的是_____。
- A) 顺序存储结构的存储一定是连续的, 链式存储结构的存储空间不一定是连续的
B) 顺序存储结构只针对线性结构, 链式存储结构只针对非线性结构
C) 顺序存储结构能存储有序表, 链式存储结构不能存储有序表
D) 链式存储结构比顺序存储结构节省存储空间
- (5) 数据流图中带有箭头的线段表示的是_____。
- A) 控制流 B) 事件驱动 C) 模块调用 D) 数据流
- (6) 软件开发中, 需求分析阶段可以使用的工具是_____。
- A) N-S 图 B) DFD 图 C) PAD 图 D) 程序流程图
- (7) 在面向对象方法中, 不属于“对象”基本特点的是_____。
- A) 一致性 B) 分类性 C) 多态性 D) 标识唯一性
- (8) 一间宿舍可住多个学生, 则实体宿舍和学生之间的联系是_____。
- A) 一对一 B) 一对多 C) 多对一 D) 多对多
- (9) 在数据管理技术发展的三个阶段中, 数据共享最好的是_____。
- A) 人工管理阶段 B) 文件系统阶段 C) 数据库系统阶段 D) 三个阶段相同
- (10) 有三个关系 R、S 和 T 如下:

R		S		T		
A	B	B	C	A	B	C
m	1	1	3	m	1	3
n	2	3	5			

由关系 R 和 S 通过进行运算得到关系 T, 则所使用的运算为_____。

- A) 笛卡儿积 B) 交 C) 并 D) 自然连接

(11) 以下叙述中正确的是_____。

- A) C 程序的基本组成单位是语句 B) C 程序中的每一行只能写一条语句
C) 简单 C 语句必须以分号结束 D) C 语言必须在一行内写完

(12) 计算机能直接执行的程序是_____。

- A) 源程序 B) 目标程序 C) 汇编程序 D) 可执行程序

(13) 以下选项中不能作为 C 语言合法常量的是_____。

- A) 'cd' B) 0.1e+6 C) "a" D) '\011'

(14) 以下选项中正确的定义语句是_____。

- A) double a; b; B) double a=b=7;
C) double a=7, b=7; D) double, a, b;

(15) 以下不能正确表示代数式 $\frac{2ab}{cd}$ 的 C 语言表达是_____。

- A) 2*a*b/c/d B) a*b/c/d*2 C) a/c/d*b*2 D) 2*a*b/c*d

(16) C 源程序中不能表示的数制是_____。

- A) 二进制 B) 八进制 C) 十进制 D) 十六进制

(17) 如有表达式 (w)? (-x): (++y), 则其中与 w 等价的表达式是_____。

- A) w==1 B) w==0 C) w!=1 D) w!=0

(18) 执行以下程序段后, w 的值为_____。

```
int w = 'A', x = 14, y = 15;
```

```
w = ((x | y) && (w < 'a'));
```

- A) -1 B) NULL C) 1 D) 0

(19) 若变量已正确定义为 int 型, 要通过语句 scanf("%d,%d,%d",&a,&b,&c); 给 a 赋值 1、给 b 赋值 2、给 c 赋值 3, 以下输入形式中错误的是 (u 代表一个空格) _____。

- A) uu1,2,3<回车> B) 1u2u3<回车>
C) 1,uuu2,uuu3 D) 1,2,3<回车>

(20) 有以下程序段:

```
int a, b, c;
```

```
a = 10; b = 50; c = 30;
```

```
if(a > b) a = b, b = c; c = a;
```

```
printf("a = %d b = %d c = %d", a, b, c);
```

程序的输出结果是_____。

- A) a=10 b=50 c=10 B) a=10 b=50 c=30
C) a=10 b=30 c=10 D) a=10 b=30 c=50

(21) 若有定义语句: int m[] = { 5, 4, 3, 2, 1 }, i = 4; , 则下面对 m 数组元素的引用中错误的是_____。

- A) m[--i] B) m[2*2] C) m[m[0]] D) m[m[i]]

(22) 下面的函数调用语句中 func 函数的实参个数是_____。

```
func(f2(v1, v2), (v3, v4, v5), (v6, max(v7, v8)));
```

- A) 3 B) 4 C) 5 D) 8

(23) 若有定义语句: double x[5] = { 1.0, 2.0, 3.0, 4.0, 5.0 }, *p = x; , 则错误引用 x 数组元素的是_____。

- A) *p B) x[5] C) *(p+1) D) *x

(24) 若有定义语句: char s[10] = "1234567\0\0", 则 strlen(s) 的值是_____。

A) 7

B) 8

C) 9

D) 10+

(25) 以下叙述中错误的是_____。

A) 用户定义的函数中可以没有 return 语句

B) 用户定义的函数中可以有多条 return 语句,以便可以调用一次返回多个函数值

C) 用户定义的函数中若没有 return 语句,则应当定义函数为 void 类型

D) 函数的 return 语句中可以没有表达式

(26) 以下关于宏的叙述中正确的是_____。

A) 宏名必须用大写字母表示

B) 宏定义必须位于源程序中所有语句之间

C) 宏替换没有数据类型限制

D) 宏调用比函数调用耗费时间

(27) 有以下程序:

#include <stdio.h>

main()

{

int i, j;

for(i=3; i>=1; i--)

{

for(j=1; j<=2; j++) printf("%d", i+j);

printf("\n");

}

}

程序运行的结果是_____。

A) 2 3 4

B) 4 3 2

C) 2 3

D) 4 5

3 4 5

5 4 3

3 4

3 4, 3 4, 3 4

2 3

(28) 有以下程序:

#include <stdio.h>

main()

{

int x=1, y=2, z=3;

if(x>y)

if(y<z) printf("%d", ++z);

else printf("%d", ++y);

printf("%d\n", x++);

}

程序运行的结果是_____。

A) 3 3 1

B) 4 1

C) 2

D) 1

(29) 有以下程序:

#include <stdio.h>

main()

{

int i=5;

do

{

if(i%3==1)

if(i%5==2)

{ printf("%d", i); break; }

- 由关系式 $i++$; 先行运算得到关系式 $i \neq 0$ 再应用
- A) } while(i!=0);
- (11) 以下 `printf("\n");` 语句
- C) 每调用一次函数 `fun` 就输出一个数, 同时 `main` 函数中 `fun` 函数调用次数为 4 次
- 程序运行的结果是 。
- (12) A) *7 B) *3*5 C) /*5 D) *2*6
- (30) 有以下程序:
- ```
#include<stdio.h>
int fun(int a,int b)
{
 if(b==0) return a;
 else return(fun(--a,--b));
}
main()
{
 printf("%d\n",fun(4,2));
}
```
- 程序运行的结果是       。
- (31) 有以下程序:
- ```
#include<stdio.h>
#include<stdlib.h>
int fun(int n)
{
    int *p;
    p=(int*)malloc(sizeof(int));
    *p=n; return *p;
}
main()
{
    int a;
    a=fun(10); printf("%d\n",a+fun(10));
}
```
- 程序运行的结果是 。
- A) 0 B) 10 C) 20 D) 出错
- (32) 有以下程序:
- ```
#include<stdio.h>
void fun(int a, int b)
{
 int t;
 t=a; a=b; b=t;
}
main()
{
 int c[10]={1,2,3,4,5,6,7,8,9,0},i;
 for(i=0;i<10;i+=2) fun(c[i], c[i+1]);
 for(i=0;i<10;i++) printf("%d,", c[i]);
 printf("\n");
}
```
- 程序运行的结果是       。
- A) 1,2,3,4,5,6,7,8,9,0      B) 2,1,4,3,6,5,8,7,0,9  
C) 0,9,8,7,6,5,4,3,2,1      D) 0,1,2,3,4,5,6,7,8,9

(33) 有以下程序:

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

```
struct st
```

```
{ int x,y; } data[2] = {1,10,2,20};
```

```
main()
```

```
{ struct st *p = data;
```

```
printf("%d,",p->y);printf("%d\n",(++p)->x);
```

```
}
```

程序运行的结果是\_\_\_\_\_。

A) 10,1

B) 20,1

C) 10,2

D) 20,2

(34) 有以下程序:

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

```
void fun(int a[],int n)
```

```
{ int i,t;
```

```
for(i=0;i<n/2;i++){t=a[i];a[i]=a[n-1-i];a[n-1-i]=t;}
```

```
}
```

```
main()
```

```
{ int k[10] = {1,2,3,4,5,6,7,8,9,10};
```

```
fun(k,5);
```

```
for(i=2;i<8;i++) printf("%d",k[i]);
```

```
printf("\n");
```

```
}
```

程序运行的结果是\_\_\_\_\_。

A) 3456787

B) 876543

C) 1098765

D) 321678

(35) 有以下程序:

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

```
#define N 4
```

```
void fun(int a[][N],int b[])
```

```
{ int i;
```

```
for(i=0;i<N;i++) b[i]=a[i][i];
```

```
}
```

```
main()
```

```
{ int x[][N] = {{1,2,3},{4},{5,6,7,8},{9,10}},y[N],i;
```

```
fun(x,y);
```

```
for(i=0;i<N;i++) printf("%d",y[i]);
```

```
printf("\n");
```

```
}
```

程序运行的结果是\_\_\_\_\_。

A) 1,2,3,4

B) 1,0,7,0

C) 1,4,5,9

D) 3,4,8,0

(36) 有以下程序:

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

```
int fun(int (*s)[4],int n,int k)
```

```
{ int m,i;
```

```
m=s[0][k];
```

```
for(i=1;i<n;i++) if(s[i][k]>m) m=s[i][k];
```

```

 return m;
}

main()
{
 int a[4][4] = {{1,2,3,4},{11,12,13,14},{21,22,23,24},{31,32,33,34}};
 printf("%d\n",fun(a,4,0));
}

```

(30) 程序运行的结果是\_\_\_\_\_。

- A) 4      B) 34      C) 31      D) 32

(37) 有以下程序:

```

#include<stdio.h>

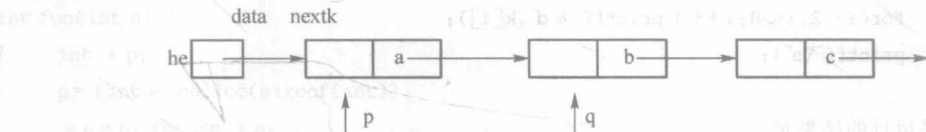
main()
{
 struct STU { char name[9]; char sex; double score[2]; };
 struct STU a = {"Zhao", 'm', 85.0, 90.0}, b = {"Qian", 'f', 95.0, 92.0};
 b = a;
 printf("%s, %c, %2.0f, %2.0f\n", b.name, b.sex, b.score[0], b.score[1]);
}

```

(37) 程序运行的结果是\_\_\_\_\_。

- A) Qian,f,95,92      B) Qian,f,85,90      C) Zhao,f,95,92      D) Zhao,m,85,90

(38) 假定已建立以下数据链表结构,且指针 p 和 q 已指向如下图所示的结点:



则以下选项中可将 q 所指结点从链表中删除并释放该结点的语句是\_\_\_\_\_。

- A) (\*p).next = (\*q).next; free(p);      B) b = q->next; free(q);  
C) p = q; free(q);      D) p->next = q->next; free(q);

(39) 有以下程序:

```

#include<stdio.h>

main()
{
 char a = 4;
 printf("%d\n", a = a << 1);
}

```

程序运行的结果是\_\_\_\_\_。

- A) 40      B) 16      C) 8      D) 4

(40) 有以下程序:

```

#include<stdio.h>

main()
{
 FILE *pf;
 char *s1 = "China", *s2 = "Beijing";
 pf = fopen("abc.dat", "wb+");
 fwrite(s2, 7, 1, pf);
 rewind(pf); /* 文件位置指针回到文件开头 */
 fwrite(s1, 5, 1, pf);
 fclose(pf);
}

```

以上程序执行后 abc.dat 文件的内容是\_\_\_\_\_。

A) China

B) Chinang

C) ChinaBeijing

D) BeijingChina

## 二、程序填空题(共 18 分)

N 个有序整数数列已放在一维数组中,给定下列程序中,函数 fun( ) 的功能是:利用折半查找法查找整数 m 在数组中的位置。若找到,则返回其下标值;反之,则返回“Not be found!”。

折半查找法的基本算法是:每次查找前先确定数组中待确定的范围:low 和 high( $low < high$ ),然后把 m 与中间位置(mid)中元素的值进行比较。如果 m 的值大于中间位置元素中的值,则下一次的查找范围放在中间位置之后的元素中;反之,下次查找范围落在中间位置之前的元素中。直到  $low > high$ ,查找结束。

注意:部分源程序给出如下。

请勿改动主函数 main 和其他函数中的任何内容,仅在函数 fun 的横线上填入所编写的若干表达式或语句。

### 【试题源程序】

```
#include<stdio.h>
#define N 10
int fun(int a[],int m)
{
 int low=0,high=N-1,mid;
 while(low<=high)
 {
 mid= 【1】 ;
 if(m<a[mid])
 high= 【2】 ;
 else
 if(m>a[mid])
 low=mid+1;
 else
 return(mid);
 }
 【3】 (-1);
}
main()
{
 int i, a[N]={-3,4,7,9,13,24,67,89,100,180},k,m;
 printf("a 数组中的数据如下:");
 for(i=0;i<N;i++);
 printf("%d",a[i]);
 printf("Enter m:");
 scanf("%d",&m);
 k=fun(a,m);
 if(k>=0)
 printf("m = %d,index = %d\n",m,k);
 else
 printf("Not be found\n");
}
```

## 三、程序修改题(共 18 分)

下列给定程序中,是建立一个带头结点的单向链表,并用随机函数为各结点数据域赋值。函数 fun 的作用

是求出单向链表结点(不包括头结点)数据域中的最大值,并且作为函数值返回。

请改正程序指定部位的错误,使它能得到正确结果。

注意:不要改动 main 函数,不得增行或删行,也不得更改程序的结构。

【试题源程序】

```
#include<stdio.h>
#include<stdlib.h>
typedef struct aa
{
 int data;
 struct aa * next;
} NODE;
fun (NODE * h)
{
 int max = -1;
 NODE * p;
 /***** found *****/
 p = h;
 while(p)
 {
 if(p->data>max)
 max = p->data;
 /***** found *****/
 p = p->next;
 }
 return max;
}
outresult(int s, FILE * pf)
{
 fprintf(pf, "\nThe max in link : %d\n", s);
}
NODE * creatlink(int n, int m)
{
 NODE * h, * p, * s, * q;
 int i, x;
 h = p = (NODE *) malloc(sizeof(NODE));
 h->data = 9999;
 for(i=1; i<=n; i++)
 {
 s = (NODE *) malloc(sizeof(NODE));
 s->data = rand() % m; s->next = p->next;
 p->next = s; p = p->next;
 }
 p->next = NULL;
 return h;
}
outlink(NODE * h, FILE * pf)
```



```

NODE *p;
p = h->next;
fprintf(pf, "\nTHE LIST : \n\n HEAD ~");
while(p)
{
 fprintf(pf, "~> %d ~", p->data); p = p->next;
}
fprintf(pf, "\n~");
}

main()
{
 NODE *head; int m;
 head = creatlink(12, 100);
 outlink(head, stdout);
 m = fun(head);
 printf("\nTHE RESULT : \n~");
 outresult(m, stdout);
}

```

#### 四、程序设计题(共 24 分)

请编写函数 fun(), 它的功能是: 判断字符串是否是回文? 若是, 函数返回 1, 主函数中输出: YES, 否则返回 0, 主函数输出 NO。回文是指顺读和倒读都一样的字符串。

例如, 字符串 LEVEL 是回文, 而字符串 12312 就不是回文。

注意: 部分原程序给出如下。

请勿改动主函数 main 和其他函数中的任何内容, 仅在函数 fun 的花括号中填入所编写的若干语句。

##### 【试题源程序】

```

#include<stdio.h>
#define N 80
int fun(char *str)
{
}

main()
{
 char s[N];
 printf("Enter a string: "); gets(s);
 printf("\n\n"); puts(s);
 if(fun(s))
 printf("YES\n");
 else
 printf("NO\n");
 NONO();
}
NONO()
{
}

```

```

/* 请在此函数内打开文件,输入测试数据,调用 fun 函数,输出数据,关闭文件。*/
FILE *rf, *wf;
int i; char s[N];
rf = fopen("K:\\k06\\24010001\\in.dat", "r");
wf = fopen("K:\\k06\\24010001\\out.dat", "w");
for(i=0; i<10; i++) {
 fscanf(rf, "%s", s);
 if(fun(s))
 fprintf(wf, "%s YES\n", s);
 else
 fprintf(wf, "%s NO\n", s);
}
fclose(rf);
fclose(wf);
}

```



## 试卷答案解析

### 一、选择题

(1) 答案: B

★ 解析: 栈是按照“先进后出”的原则组织数据的,入栈的顺序为 12345ABCDE, 1 为栈底元素最后出栈, E 为栈顶元素最先出栈, 因此出栈的顺序为 EDCBA54321。

(2) 答案: D

★ 解析: 循环队列是将队列存储空间的最后一个位置绕到第一个位置, 形成逻辑上的环形空间。循环队列仍然是顺序存储结构, 是队列常采用的形式, 因此选项 A 错误。在循环队列中, 用队尾指针 rear 指向队列中的队尾元素, 用队头指针 front 指向队列排头元素的前一个位置。循环队列中的元素是动态变化的, 每进行一次入队运算, 对尾指针就进一; 每进行一次出队运算, 队头指针就进一。可见由队头指针和队尾指针一起反映队列中元素的动态变化情况, 因此选项 B、C 是错误的。从队头指针 front 指向的后一个位置直到队尾指针 rear 指向的位置之间所有的元素均为队列中的元素, 因此选项 D 是正确的。

(3) 答案: C

★ 解析: 最糟糕的情况应该也是比较到线性表最后一个值, 也没有查找到所需要的值, 那么从线性表的第 0 个值开始比较, 每次取出一个值比较, 不符合, 再取下一个值, 依次比较, 一直到最后一个, 那么长度为 N, 就需要比较 n 次。

(4) 答案: A

★ 解析: 在顺序存储结构中所有元素所占的存储空间是连续的, 而在链式存储结构中, 存储数据结构的存储空间可以不连续, 因此选项 A 是正确的。线性表在计算机中的存放可以采用顺序存储结构, 也可采用链式存储结构, 顺序存储结构和链式存储结构都是既可用于线性结构, 也可以用于非线性结构, 因此选项 B、C 是错误的。采用链式存

储结构, 不仅要存储元素的值, 元素间的逻辑关系还需要通过附设的指针字段来表示, 因此, 链式存储结构需要更多的存储空间。

(5) 答案: D

★ 解析: 数据流图的基本符号的意思: ①矩形表示数据的外部实体; ②圆角的矩形表示变换数据的处理逻辑; ③少右面的边矩形表示数据的存储; ④箭头表示数据流。

(6) 答案: B

★ 解析: 结构化解析方法是常见的需求分析方法之一, 它是结构化程序设计理论在软件需求解析阶段的运用, 数据流图(Data Flow Diagram, DFD)是结构化解析常用的工具之一。数据字典、判定树和判定表也是常用的结构化分析工具。程序流程图、N-S 图、PAD 图是详细设计过程中常用的图形工具。

(7) 答案: A

★ 解析: 对象的基本特点有: 标识唯一性、分类性、多态性、封装性和模块独立性好。标识唯一性是指对象是可区分的; 分类性是指可将具有相同属性和操作的对象抽象成类; 多态性是指同一个操作可以是不同对象的行为; 封装性是指对象的内部对外不可见, 在外面不可直接使用对象的处理能力, 也不能直接修改其内部状态。

(8) 答案: B

★ 解析: 一间宿舍对应多个学生, 则宿舍和学生之间的联系是一对多, 学生和宿舍的联系是多对一。

(9) 答案: C

★ 解析: 在人工管理阶段, 数据不保存, 没有对数据进行管理的软件, 数据不能共享; 文件系统是数据库系统发展的初级阶段, 提供了简单的数据共享与数据管理能力, 但它不能提供完整的、统一的、管理和数据共享的能力。数据

库系统阶段实现了数据共享,具有数据的集成性,这使得数据可为多个应用所共享,数据的共享自身又可极大地减少数据冗余性。

(10) 答案:D

解析: 交和并运算要求关系 R 和 S 具有相同的属性,因此可以排除选项 B、D。R 与 S 的笛卡儿积是一个 4 元关系,有 4 个元素,因此 A 错误。R 与 S 有公共的属性 B,可通过公共属性的相等值进行链接,这就是自然连接,可见选项 C 是正确的。

(11) 答案:C

解析: 本题考查 C 语言基本概念,C 程序的基本组成单位是函数,故选项 A 错。C 程序一行能写多条语句,也可以将一条语句分几行书写,故 B、C 选项错,在 C 语言中每条语句必须以分号结束。故选 D。

(12) 答案:D

解析: 源程序是由程序设计语言构成的指令序列,需要有特定的软件去执行。C 语言是一种高级语言,C 语言源程序经过编译器编译后,生成 .obj 文件然后经过链接生成 .exe 文件,所以源程序是不能直接执行的。

(13) 答案:A

解析: 常量包括整型常量、实型常量、字符常量和字符串常量等。选项 B 中  $0.1e+6$  是指数形式的常量是合法的;选项 C 中 `\a` 代表非打印的转义字符,是字符串常量;选项 D 中 `\011` 是表示八进制整型常量。故选项 A 是错误的,单引号表示字符常量,不能包涵字符串,若要包含字符串需用双引号。

(14) 答案:C

解析: 同一类型变量的定义时,不同变量之间需要用“,”分隔,选项 A 中,a 和 b 之间缺少逗号而用分号是不正确的,分号是语句结束标志,因此,选项 A 是错误的;定义变量时同时为变量初始化赋值是不能用连等,因此选项 B 错误。选项 D 是错误的,在 double 和 a 之间多以一个逗号,变量类型说明后面不应用逗号,应该用空格分隔。

(15) 解析: 答案:D

本题主要考查运算符的优先级和结合性。`*`、`/` 优先级相同所以采用左结合的方式。选项 D 应该改为: `2 * a * b / (c * d)`,因此 D 是错误的。

(16) 答案:A

解析: C 语言中整形常量可以是。十进制,八进制数,十六进制数虽然运行方式以二进制运行,但整形常量不包括二进制。

(17) 答案:D

解析: 本题主要考查关系表达式。此条件表达式的形式为 `<表达式 1>? <表达式 2>: <表达式 3>`,当表达式 1 的值为真即为非零时,求出表达式 2 的值作为结果;当表达式 1 的值为假,求出表达式 3 的值作为结果。可见表达式 w 等价于 `w!=0`。

(18) 答案:C

解析: 本题考查逻辑表达式,因为 `x=14`,`y=15` 都是非零所以都为真,因此 `x||y=1`;又因为 `'A'+32='a'`,

知 `w<'a'` 为真,因此 `(x||y)&&(w<'a')` 的值为 1。

(19) 答案:B

解析: 在本题 scanf 函数中,每个输入格式说明之间加一个逗号作为通配符,要求除格式控制符以外的字符都要原样输入,逗号为非格式符需要原样输入,否则就不能正确读入数据。选项 B 中,前两个数据后面没有紧跟一个逗号,格式不匹配,错误。

(20) 答案:A

解析: 本题考查 if 语句,本题是一道陷阱题,`a=10`,`b=50`,则 `a<b`,if 语句条件不成立,执行 `c=a`,得到 `c=10`,a 和 b 的值不变,但是很多考生会以为 `c=a` 也是 if 语句的一部分,这是错误的,if 语句在没有 `{}` 说明时仅仅是跟随其后的第一条语句。

(21) 答案:C

解析: 数组的下标是从零开始标记,数组 m 中有 5 个元素,所以数组 m 即为,`m[0]=5`,`m[1]=4`,`m[2]=3`,`m[3]=2`,`m[4]=1`。C 中的数组元素为 `m[5]` 但是数组申请的空间是 `0~4` 可见 `m[5]` 会溢出,是错误的引用。

(22) 答案:A

解析: 本题考查函数调用,函数调用中,实参的为多个时,实参之间用逗号分隔。本题中的实参有 3 个,为 `f2(v1, v2)`、`(v3, v4, v5)`、`(v6, max(v7, v8))`。其中 `f2(v1, v2)` 是函数做参数,(`v3, v4, v5`) 是逗号表达式做参数,(`v6, max(v7, v8)`) 是逗号与函数相结合做参数。

(23) 答案:B

解析: 对一维数组元素可以采用多种方式。直接引用,数组名[下标],数组的大小为 n 时,下标的取值范围为 `0~(n-1)`。本题中数组 x 的下标为 `0~4`,`x[5]` 溢出,因此选项 B 错误。通过指针引用一维数组元素,指针 p 指向该数组,所以 `*p=x[0]` 指针 p+1 指向数组 x 的第二个元素的地址,`*(p+1)=x[1]` 所以 C 选项正确,x 为 x 数组元素的首地址,x 的值等于 `&x[0]`,`*x=x[0]`,因此选项 D 正确,故选择 B。

(24) 答案:A

解析: 在 C 语言中,字符串可以存放数字字符串,但规定以字符 `'\0'` 作为字符串结束标识。所以字符数组的长度并不是字符串的长度,系统字符串 `"1234567\0\0"` 与 `"1234567"` 等效,所以字符串长度是有效字符数组的长度即为 7。

(25) 答案:B

解析: 定义函数有两种一种是 void 函数,可以没有返回 return 语句,若有就不能有任何表达式;另一种是指定返回类型,则必须有一个返回语句,在一个函数内,可以根据需要在多处出现 return 语句。但无论函数体内有多少个 return 语句,return 语句只返回一个函数值。故选 B。

(26) 答案:C

解析: 用作宏名的标识符通常习惯用大写字母表示,宏名可以是任何合法的 C 语言标识符,可见选项 A 是错误的。在 C 语言中宏定义可以根据需要写在任何位置,因此选项 B 错误。宏调用是在编译前由编译预处理程序完成

的,不占运行的时间,因此选项 D 错误。

(27) 答案:D

★ 解析:本题主要考查 for 循环语句的嵌套。外层主循环 i 从 3 减到 1 执行了 3 次,嵌套的循环变量 j 从 1 增到 2 句每轮执行 2 次,每次输出 i+j 的值,即为 3+1,3+2,2+1,2+2,1+1,1+2 但注意每次内循环结束要换行。

(28) 答案:D

★ 解析:本题主要考查嵌套的 if-else 语句。C 语言规定:else 子句总是与最前面最近的不带 else 的 if 相结合,与书写格式无关。本题中的 else 语句与第二个 if 语句配对。程序首先判断  $x > y$  不成立,直接退出 if 语句,执行  $\text{printf}("%d\n", x++)$ ; 注意  $x++$  是先输出后做 ++ 运算,故首先输出 x 的值 1,然后将 x 的值加 1。

(29) 答案:A

★ 解析:本题主要考查 do-while 语句。在 do-while 循环中,总是先执行后判断,即至少会执行一次。在循环中,如果满足  $(i \% 3 == 1) \&\& (i \% 5 == 2)$  那么输入 i 的值,退出循环。

(30) 答案:B

★ 解析:本题主要考查函数的递归调用。fun 函数的功能是:当 b 的值为零时,返回此时 a 的值;否则,返回 fun( $a-1, b-1$ ),即将 a 和 b 的值分别减 1 后返回 fun(a, b)。从这里可以看出 b 不断递减时, a 也不断递减直到 b 为零。执行 fun(4, 2), b=2, 返回 f(3, 1), 此时 b=1, 返回 f(2, 0), b=0, 返回 a 的值 2。

(31) 答案:C

★ 解析:函数 fun 的功能是:动态分配 int 型大小的内存空间,指针 p 指向该存储区的首地址,并将它初始化为参数 n 的值,然后返回这个值,函数将传递给形参的变量保存在该存储区。 $a = \text{fun}(10) = 10$ , 则  $a + \text{fun}(10) = 20$ 。

(32) 答案:A

★ 解析:本题考查参数的传递在 C 语言中。在 C 语言中函数参数的传递是“传值”的,即形参和实参是两个没有关系的变量,函数 fun 虽然看似交换了参数值但是只交换了形参的值,但结果不会传递给实参。因此数组 c 没发生变化,所以原样输出。

(33) 答案:C

★ 解析:本题主要考查结构体指针。题目首先定义结构体 st 的同时定义结构体数组 data,可知  $\text{data}[0] = \{1, 10\}$ ,  $\text{data}[1] = \{2, 20\}$ 。结构体指针 p 指向数组的首地址,指向  $\text{data}[0]$ ,  $++p$  则指向数组的第二个元素  $\text{data}[1]$ , 然后输出  $p \rightarrow y$ ,  $(++p) \rightarrow x$  即输出  $\text{data}[0].y$  和  $\text{data}[1].x$  所以分别为 10 和 2。

(34) 答案:D

二、程序填空题

★ 解析:

填空 1:根据题目的意思,这里应该是确定折半查找的中间位置,所以很明显应该填  $(\text{low} + \text{high}) / 2$ , 注意这个式子返回的是整型数据,即如果分子为 7 则结果为 3。

填空 2:根据题目的意思,中间的元素值大时应该选择前半段进行下次查找,所以应该把 mid 前一位的下标赋值给

★ 解析:一维数组名作为实参传递时,由于数组名本身是一个地址值,对应的形参相当于一个指针变量,在函数中可以通过指针变量来引用调用函数中对应的数组元素,从而达到对调用函数中的数组元素进行操作而改变其中的值。本题中函数 fun 的作用是将数组 a 的前 n 个元素进行逆转, fun(k, 5) 的结果是数组 k 的前五个元素 1、2、3、4、5 变为 5、4、3、2、1。置换后,输出数组中第 3 个~第 8 个元素。

(35) 答案:B

★ 解析:由二维数组 x 的定义知,数组 x 为 4 行 4 列。fun 函数中 for 循环的作用是将二维数组前 4 行的对角线元素付给  $b[0]$ ,  $b[1]$ ,  $b[2]$ ,  $b[3]$  在主函数中定义二维数组为  $\{\{1, 2, 3\} \{4\} \{5, 6, 7, 8\} \{9, 10\}\}$  所以可以看出对角线元素为 1, 0, 7, 0。故选 B。

(36) 答案:C

★ 解析:本题主要考查通过数组指针引用二维数组的方法。函数 fun 的作用是遍历  $s[1][k] \sim s[n-1][k]$  之中的数,找出最大的元素。所以依据题意得出最大数为 31。

(37) 答案:D

★ 解析:对结构体变量赋初值时,在 C 语言中相同类型变量是可以通过等号直接赋值,它会将每个成员在结构体中的顺序一一对应赋初值,可知,  $a.name = "Zhao"$ ,  $a.sex = 'f'$ ,  $a.score[0] = 85.0$ ,  $a.score[1] = 90.0$ ,  $b.name = "Qian"$ ,  $b.sex = 'm'$ ,  $b.score[0] = 95.0$ ,  $b.score[1] = 92.0$ 。输出函数中要求输出的浮点有效位为 2 位,因此选项 D 是正确的。

(38) 答案:D

★ 解析:本题主要考查由结构体变量构成的单向链表的操作。要删除结点 q, 首先由 q 的上一个结点 p 的指针域指向 q 的指针域所指向的结点  $p \rightarrow next = q \rightarrow next$ , 然后释放结点 q, free(q) 所以只有 D 选项正确。

(39) 答案:C

★ 解析:本题主要考查左移运算。左移一位相当于原数乘以二,右移相当于除以二,  $a < 1$  结果为  $4 * 2 = 8$ 。没有超过 char 的类型表示范围,所以输出 8。

(40) 答案:B

★ 解析:首先定义了一个文件指针 pf, 然后执行  $\text{fopen}("abc.dat", "wb+")$ ; 建立可读写的二进制文件 abc.dat, 然后执行语句  $\text{fwrite}(s2, 7, 1, pf)$ ; 将 s2 的前 7 \* 1 个字符的内容写入 pf 中, 现在应该为 Beijing 然后由  $\text{rewind}(pf)$ ; 将文件位置指针回到文件开头, 执行语句  $\text{fwrite}(s1, 5, 1, pf)$ ; 从文件的开头位置写入 s1 中前五个字符, 所以结果为 Chinang。