

目 录

2013 年 9 月全国计算机等级考试	笔试真题	1
无纸化二级 C 语言	上机真题	9
无纸化题库试题	笔试真题答案解析	12
无纸化题库试题答案解析	上机真题答案解析	15
2013 年 3 月全国计算机等级考试	2010 年 9 月全国计算机等级考试二级 C 语言	(共 16 页)
无纸化二级 C 语言	笔试真题	1
无纸化题库试题	上机真题	9
无纸化题库试题答案解析	笔试真题答案解析	12
2012 年 9 月全国计算机等级考试	上机真题答案解析	15
无纸化二级 C 语言	2010 年 3 月全国计算机等级考试二级 C 语言	(共 16 页)
无纸化题库试题	笔试真题	1
无纸化题库试题答案解析	上机真题	10
2012 年 9 月全国计算机等级考试二级 C 语言	笔试真题答案解析	13
笔试真题	上机真题答案解析	15
上机真题	2009 年 9 月全国计算机等级考试二级 C 语言	(共 16 页)
笔试真题答案解析	笔试真题	1
上机真题答案解析	上机真题	10
2012 年 3 月全国计算机等级考试二级 C 语言	笔试真题答案解析	12
笔试真题	上机真题答案解析	15
上机真题	2009 年 3 月全国计算机等级考试二级 C 语言	(共 16 页)
笔试真题答案解析	笔试真题	1
上机真题答案解析	上机真题	9
2011 年 9 月全国计算机等级考试二级 C 语言	笔试真题答案解析	11
笔试真题	上机真题答案解析	15
上机真题	2008 年 4 月 ~ 2008 年 9 月笔试真题及答案解析	(见光盘)
笔试真题答案解析	无纸化真题库(共 105 套)	(见光盘)
上机真题答案解析	无纸化操作题高频考点速记	(另分册)
2011 年 3 月全国计算机等级考试二级 C 语言		

注:解析中【真题链接】部分所涉及题目均为本书配套光盘软件中的“操作题单项训练”模块中对应的题目。

2013 年 9 月全国计算机等级考试无纸化二级 C 语言

无纸化题库试题

(考试时间 120 分钟,满分 100 分)

一、选择题(每小题 1 分,共 40 分)

- (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)循环队列有队头和队尾两个指针,因此,循环队列是非线性结构
- B)在循环队列中,只需要队头指针就能反映队列中元素的动态变化情况
- C)在循环队列中,只需要队尾指针就能反映队列中元素的动态变化情况
- D)循环队列中元素的个数是由队头指针和队尾指针共同决定的
- (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)在软件开发中,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)下列叙述中错误的是()。

- A)C 程序可以由多个程序文件组成
- B)一个 C 语言程序只能实现一种算法
- C)C 程序可以由一个或多个函数组成
- D)一个 C 函数可以单独作为一个 C 程序文件存在

(12)以下选项中,能用作数据常量的是()。

- A)115L
- B)0118
- C)1.5e1.5
- D)o115

(13)按照 C 语言规定的用户标识符命名规则,不能出现在标识符中的是()。

- A)大写字母
- B)下划线
- C)数字字符
- D)连接符

(14)设变量已正确定义并赋值,以下正确的表达式是()。

- A)x = y + z + 5, ++y
- B)int(15.8%5)
- C)x = y * 5 = x + z
- D)x = 25%5.0

(15)设有定义:int x=2;;以下表达式中,值不为 6 的是()。

- A)2 * x, x += 2
- B)x ++ , 2 * x
- C)x *= (1 + x)
- D)x *= x + 1

(16)有以下程序:

```
#include <stdio.h>

main()
{
    int x, y, z;
    x = y = 1;
    z = x ++ , y ++ , ++ y;
    printf( "%d,%d,%d\n" ,x,y,z);
}
```

程序运行后的输出结果是()。

- A)2,3,3
- B)2,3,2
- C)2,3,1
- D)2,2,1

(17)有以下程序:

```
#include <stdio.h>

main()
{
    char c1,c2;
    c1 = 'A' + '8' - '4';
    c2 = 'A' + '8' - '5';
    printf( " %c,%d\n" ,c1,c2);
}
```

已知字母 A 的 ASCII 码为 65,程序运行后的输出结果是()。

- A)E,68
- B)D,69
- C)E,D
- D)输出无定值

(18)若有定义 int x,y; 并已正确给变量赋值,则以下选项中与表达式(x-y)? (x++) : (y++)中的条件表达式 (x-y) 等价的是()。

- A)(x-y<0||x-y>0)
- B)(x-y<0)
- C)(x-y>0)
- D)(x-y == 0)

(19)有以下程序:

```
#include <stdio.h>

main()
{
    int x = 1, y = 0;
    if (! x) y ++ ;
    else if (x == 0)
        if (x) y ++ = 2;
```

```

        else y + =3;
    printf( "%d\n" , y );
}

```

程序运行后的输出结果是()。

- A)3 B)2 C)1 D)0

(20)若有定义:float x=1.5; int a=1,b=3,c=2; 则正确的 switch 语句是()。

- A)switch(a+b) B)switch((int)x);
- ```

 { case 1: printf(" * \n");
 case 2+1: printf(" * * \n"); }

```
- C)switch(x)                      D)switch(a+b)
- ```

    { case 1.0: printf( " * \n" );
      case 2.0: printf( " * * \n" ); }

```
- ```

 { case 1: printf(" * \n");
 case 2: printf(" * * \n"); }

```
- case c: printf( " \* \* \n" ); }

(21)有以下程序:

```

#include <stdio.h>
main()
{
 int y=9;
 for(; y>0; y--)
 if(y%3==0) printf("%d", --y);
}

```

程序的运行结果是( )。

- A)852                      B)963                      C)741                      D)875421

(22)有以下程序:

```

#include <stdio.h>
main()
{
 int i,j,m=1;
 for(i=1;i<3;i++)
 for(j=3;j>0;j--)
 if(i*j>3) break;
 m*=i*j;
 }
 }
 printf("m=%d\n",m);
}

```

程序运行后的输出结果是( )。

- A)m=4                      B)m=2                      C)m=6                      D)m=5

(23)有以下程序:

```

#include <stdio.h>
int fun (int x, int y)
{
 if (x!=y) return ((x+y)/2);
 else return (x);
}
main()
{
 int a=4, b=5, c=6;
}

```

```
printf("%d\n" , fun(2 * a, fun(b, c)));
}
```

程序运行后的输出结果是( )。

A)6

B)3

C)8

D)12

(24)有以下程序:

```
#include <stdio.h>
int add(int a,int b){ return (a+b); }
main()
{ int k, (*f) (), a=5,b=10;
 f = add;
 ...
}
```

则以下函数调用语句错误的是( )。

A)k=f(a,b);

B)k=add(a,b);

C)k=( \*f )(a,b);

D)k= \*f(a,b);

(25)有以下程序:

```
#include <stdio.h>
main()
{ int n, *p=NULL;
 *p=&n;
 printf("Input n:"); scanf("%d",&p); printf("output n:"); printf("%d\n",p);
}
```

该程序试图通过指针 p 为变量 n 读入数据并输出,但程序有多处错误,以下语句正确的是( )。

A)int n, \*p=NULL;

B)\*p=&n;

C)scanf("%d",&p)

D)printf("%d\n",p);

(26)下列定义数组的语句中,正确的是( )。

A)#define N 10

B)int N=10;

int x[N];

int x[N];

C)int x[0...10];

D)int x[ ];

(27)有以下程序:

```
#include <stdio.h>
main()
{ int a[] = {2,3,5,4},i;
 for(i=0;i<4;i++)
 switch(i%2)
 { case 0: switch(a[i]%2)
 { case 0:a[i]++;break;
 case 1:a[i]--;
 } break;
 case 1:a[i]=0;
 }
 for(i=0;i<4;i++) printf("%d ",a[i]); printf("\n");
}
```

程序运行后的输出结果是( )。

A)0 3 0 4

B)2 0 5 0

C)3 3 4 4

D)3 0 4 0

(28)有以下程序:

```
#include <stdio.h>

main()
{
 int b[3][3] = {0,1,2,0,1,2,0,1,2}, i, j, t = 1;
 for(i = 0; i < 3; i++)
 for(j = i; j < = i; j++) t += b[i][b[j][i]];
 printf("%d\n", t);
}
```

程序运行后的输出结果是( )。

A)3

B)4

C)1

D)9

(29)以下语句中存在语法错误的是( )。

A)char ss[6][20]; ss[1] = "right?";

B)char ss()[20] = { "right?" };

C)char \*ss[6]; ss[1] = "right?";

D)char \*ss() = { "right?" };

(30)以下不能将 s 所指字符串正确复制到 t 所指存储空间的是( )。

A)do { \*t++ = \*s++; } while( \*s );

B)for(i = 0; t[i] = s[i]; i++);

C)while( \*t = \*s ) { t++; s++; }

D)for(i = 0, j = 0; t[i++] = s[j++]; );

(31)有以下程序:

```
#include <stdio.h>

void swap(char *x, char *y)
{
 char t;

 t = *x; *x = *y; *y = t;
}

main()
{
 char *s1 = "abc", *s2 = "123";
 swap(s1, s2); printf("%s,%s\n", s1, s2);
}
```

程序执行后的输出结果是( )。

A)321, cba

B)abc, 123

C)123, abc

D)1bc, a23

(32)有以下函数:

```
int fun(char *x, char *y)
{
 int n = 0;
 while ((*x == *y) && *x != '\0') { x++; y++; n++; }
 return n ;
}
```

函数的功能是( )。

A)将 y 所指字符串赋给 x 所指存储空间

B)查找 x 和 y 所指字符串中是否有 '\0'

C)统计 x 和 y 所指字符串中最前面连续相同的字符个数

D)统计 x 和 y 所指字符串中相同的字符个数

(33)有以下程序:

```
#include <stdio.h>

int fun()
```

```

{
 static int x = 1;

 x * = 2;

 return x;
}

main()
{
 int i, s = 1;

 for(i = 1; i < = 3; i + +) s * = fun();

 printf(" %d\n" , s);
}

```

程序运行后的输出结果是( )。

- A)10                      B)30                      C)0                      D)64

(34)在一个 C 源程序文件中所定义的全局变量,其作用域为( )。

- A)由具体定义位置和 extern 说明来决定范围      B)所在程序的全部范围  
C)所在函数的全部范围                      D)所在文件的全部范围

(35)以下叙述中错误的是( )。

- A)可以通过 typedef 增加新的类型  
B)可以用 typedef 将已存在的类型用一个新的名字来代表  
C)用 typedef 定义新的类型名后,原有类型名仍有效  
D)用 typedef 可以为各种类型起别名,但不能为变量起别名

(36)有以下程序:

```

#include <stdio.h>

struct S
{
 int a, b; } data[2] = {10,100,20,200};

main()
{
 struct S p = data[1];

 printf(" %d\n" , + + (p. a));
}

```

程序运行后的输出结果是( )。

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

(37)有以下程序:

```

#include <stdio.h>

#define PT 3.5 ;

#define S(x) PT * x * x ;

main()
{
 int a = 1, b = 2; printf(" %4.1f\n" , S(a + b)); }

```

程序运行后的输出结果是( )。

- A)7.5                      B)31.5  
C)程序有错无输出结果                      D)14.0

(38)有以下程序:

```

include <stdio.h>

main()
{
 unsigned char a = 2, b = 4, c = 5, d;
}

```

```
d = a | b; d &= c; printf("%d\n", d); }
```

程序运行后的输出结果是( )。

A)3

B)4

C)5

D)6

(39)有以下程序:

```
#include <stdio.h>
#include <stdlib.h>
main()
{ int *a, *b, *c;
 a = b = c = (int *) malloc(sizeof(int));
 *a = 1; *b = 2, *c = 3;
 a = b;
 printf("%d,%d,%d\n", *a, *b, *c);
}
```

程序运行后的输出结果是( )。

A)1,1,3

B)2,2,3

C)1,2,3

D)3,3,3

(40)以下叙述中正确的是( )。

A) 当对文件的读(写)操作完成之后,必须将它关闭,否则可能导致数据丢失

B) 打开一个已存在的文件并进行了写操作后,原有文件中的全部数据必定被覆盖

C) 在一个程序中当对文件进行了写操作后,必须先关闭该文件然后再打开,才能读到第1个数据

D) C语言中的文件是流式文件,因此只能顺序存取数据

## 二、程序填空题(共18分)

下列给定程序中,函数fun的功能是:计算如下公式前n项的和,并作为函数值返回。

$$S = \frac{1 \times 3}{2^2} + \frac{3 \times 5}{4^2} + \frac{5 \times 7}{6^2} + \cdots + \frac{(2 \times n - 1) \times (2 \times n + 1)}{(2 \times n)^2}$$

例如,当形参n的值为10时,函数返回值为9.612558。

请在下画线处填入正确的内容并将下画线删除,使程序得出正确的结果。

注意:部分源程序给出如下。

不得增行或删行,也不得更改程序的结构!

试题程序:

```
#include <stdio.h>
double fun(int n)
{ int i; double s, t;
 /***** found *****/
 s = 1;
 /***** found *****/
 for(i=1; i <= 2; i++)
 { t = 2.0 * i;
 /***** found *****/
 s = s + (2.0 * i - 1) * (2.0 * i + 1) / 3;
 }
 return s;
}

void main()
```



```

{ int n = -1;
 while(n<0)
 { printf("Please input (n>0): ");
 scanf("% d", &n); }
 printf("\nThe result is: % f\n", fun(n));
}

```

### 三、程序修改题(共 18 分)

下列给定程序中函数 fun 的功能是:统计 substr 所指的字符串在 str 所指的字符串中出现的次数。

例如,若字符串为 aaas lkaaas,子字符串为 as,则应输出 2。

请改正程序中的错误,使它能得出正确的结果。

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!

试题程序:

```

#include <stdio.h>

int fun (char * str, char * substr)
{ int i, j, k, num=0;
 /***** found *****/
 for(i = 0, str[i], i++)
 for(j=i, k=0;substr[k] == str[j];k++, j++)
 /***** found *****/
 If(substr[k+1] == '\0')
 { num++;
 break;
 }
 return num;
}

main()
{
 char str[80], substr[80];
 printf("Input a string:") ;
 gets(str);
 printf("Input a substring:") ;
 gets(substr);
 printf("% d\n", fun(str, substr));
}

```

### 四、程序设计题(共 24 分)

编写函数 fun,其功能是:根据以下公式求  $\pi$  的值(要求精度 0.0005,即某项小于 0.0005 时停止迭代)。

$$\frac{\pi}{2} = 1 + \frac{1}{3} + \frac{1 \times 2}{3 \times 5} + \frac{1 \times 2 \times 3}{3 \times 5 \times 7} + \frac{1 \times 2 \times 3 \times 4}{3 \times 5 \times 7 \times 9} + \cdots + \frac{1 \times 2 \times \cdots \times n}{3 \times 5 \times \cdots \times (2n+1)}$$

程序运行后,若输入精度 0.0005,则程序应输出为 3.14 …。

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 的花括号中填入你编写的若干语句。

试题程序:

```

#include <stdio.h>

```

```
#include <conio.h>
#include <math.h>
double fun(double eps)
{
 double x;
 while(fabs(sin(x)-cos(x))>eps)
 {
 x+=0.0001;
 }
 return x;
}
void main()
{
 double x;
 printf("Input eps:");
 scanf("%lf", &x);
 printf("\neps=%lf, PI=%lf\n", x, fun(x));
}
```

## 无纸化题库试题答案解析

### 一、选择题

(1)B) 【解析】栈是按照先进后出的原则组织数据,所以入栈最早的最后出栈,所以选择 B)。

(2)D) 【解析】循环队列有队头和队尾两个指针,但是循环队列仍是线性结构的,所以 A) 错误;在循环队列中只需要队头指针与队尾两个指针来共同反映队列中元素的动态变化情况,所以 B) 与 C) 错误。

(3)C) 【解析】当有序线性表为顺序存储时才能用二分法查找。可以证明的是对于长度为  $n$  的有序线性表,在最坏情况下,二分法查找只需要比较  $\log_2 n$  次,而顺序查找需要比较  $n$  次。

(4)A) 【解析】链式存储结构既可以针对线性结构也可以针对非线性结构,所以 B) 与 C) 错误。链式存储结构中每个结点都由数据域与指针域两部分组成,增加了存储空间,所以 D) 错误。

(5)D) 【解析】数据流图中带箭头的线段表示的是数据流,即沿箭头方向传送数据的通道,一般在旁边标注数据流名。

(6)B) 【解析】在需求分析阶段可以使用的工具有数据流图 (DFD)、数据字典 (DD)、判定树与判定表,所以选择 B)。

(7)A) 【解析】对象有如下一些基本特点:标识唯一性、分类性、多态性、封装性、模块独立性好。所以选择 A)。

(8)B) 【解析】因为一间宿舍可以住多个学生,即多个学生住在一个宿舍中,但一个学生只能住一间宿舍,所以实体宿舍和学生之间是一对多的关系。

(9)C) 【解析】数据管理发展至今已经历了 3 个阶段:人工管理阶段、文件系统阶段和数据库系统阶段。其中最后一个阶段结构简单,使用方便,逻辑性强,物理性少,在各方面的表现都最好,一直占据数据库领域的主导地位,所以选择 C)。

(10)D) 【解析】自然连接是一种特殊的等值连接,它要求两个关系中进行比较的分量必须是相同的属性组,并且在结果中把重复的属性列去掉,所以根据 T 关系中的有序组可知 R 与 S 进行的是自然连接操作。

(11)B) 【解析】在一个 C 语言程序中可以进行多种算法的实现,对算法的个数没有规定,所以 B) 错误。

(12)A) 【解析】A) 选项中 115L 表示 115 是长整型数据,合法。B) 选项是八进制常量的表示方法,但是在八进制中不能含有数字 8,所以 B) 错误。C) 选项中 e 后面应该是整数不能是小数 1.5,所以 C) 错误。D) 选项中八进制常量应该是数字“0”开始,而不是字母“o”开始。

(13)D) 【解析】C 语言中标识符由字母、下划线、数字组成,且开头必须是字母或下划线。所以 D) 选项中的连接符不合法。

(14)A) 【解析】B)与 D)选项中取模运算符%的左右两个操作数应均为整数,所以 B)、D)错误。C)选项中不能将  $x+y$  的值赋给表达式  $y*5$ ,所以 C)错误。

(15)A) 【解析】A)选项中逗号表达式先计算第一表达式  $2*x$ ,然后计算表达式  $x+=2$  的值,即  $x=x+2$  即 4,整个逗号表达式为第二个表达式的值 4,所以选择 A)。B)选项中首先计算逗号表达式中第一表达式  $x++$ ,此时  $x$  为 3,在执行第二个表达式  $2*x=2*3=6$ ,所以逗号表达式为第二个表达式的值 6。C)选项的赋值表达式可以表示为  $x=x*(1+x)=2*(1+2)=6$ 。D)选项中的表达式可以表示为  $x=x*(x+1)=2*3=6$ 。

(16)C) 【解析】 $z=x++,y++,++y;$ ,因为赋值运算符的优先级高于逗号运算符的优先级,所以可以将上式表示成  $(z=x++),(y++),(++y)$ 。然后从左向右先计算表达式  $z=x++$ ,因为  $x++$  先使用后自增,所以  $z$  的值为 1, $x$  的值为 2。再计算逗号表达式第二个表达式  $y++$ ,此时  $y$  的值为 2,最后计算第三个表达式  $++y$ , $y$  的值为 3。

(17)A) 【解析】本题目中字符变量  $c1$  是字符 'A' 的 ASCII 码加上 4,即 69 所对应的字符 'E'。字符变量  $c2$  是字符 'A' 的 ASCII 码加上 3,即 68 所对应的字符 'D'。但是打印输出时, $c1$  以 %c 的格式输出,所以是 E, $c2$  以 %d 的格式输出,所以是 68。

(18)A) 【解析】条件表达式:  $x = \text{表达式} 1 ? \text{表达式} 2 : \text{表达式} 3$  的含义是:先求解表达式 1,若为非 0(真),则求解表达式 2,将表达式 2 的值赋给  $x$ ;若表达式 1 的值为 0(假),则求解表达式 3,将表达式 3 的值赋给  $x$ 。在本题中与表达式 1:  $(x-y)$  等价的是  $(x-y < 0 || x-y > 0)$ 。

(19)D) 【解析】在 if else 语句中,else 总是与离它最近的 if 配对。本题目中  $x$  为 1,所以  $!x$  为 0,因此执行 else if 语句中的内容,判断  $(x=0)$  是否成立。因为  $x$  为 1,所以条件不成立,于是 else if 内部的 if...else 语句不再执行,所以  $y$  的值还是初始值 0。

(20)A) 【解析】B)选项中  $\text{switch}((\text{int})x)$ ;语句中不应该有最后的分号。 $\text{switch}(\text{expr}1)$  中的  $\text{expr}1$  不能用浮点类型或 long 类型,也不能为一个字符串,所以 C)错误。case 后面常量表达式的类型必须与 switch 后面表达式的类型一致,所以 D)错误。

(21)A) 【解析】第一次 for 循环, $y$  的值为 9, $y\%3$  的值为 0,满足条件打印  $--y$ ,即先减一后打印,所以打印 8;第二次 for 循环, $y$  的值为 7, $y\%3$  的值为 1,不执行打印语句;第三次 for 循环, $y$  的值为 6, $y\%3$  的值为 0,满足条件打印  $--y$ ,即先减一后打印,所以打印 5;第四次 for 循环, $y$  的值为 4,不满足 if 条件,不执行打印语句;第五次 for 循环, $y$  的值为 3,满足 if 条件,打印输出 2;第六次 for 循环, $y$  的值为 1,不满足条件,不执行打印语句。

(22)C) 【解析】第一次外循环  $i$  的值为 1,第一次内循环  $j$  的值为 3,不满足条件执行  $m*=i*j$ ,即  $m$  的值为 3;第二次  $j$  的值为 2,不满足条件执行  $m*=i*j$ ,即  $m$  的值为 6;第三次  $j$  的值为 1,不满足条件执行  $m*=i*j$ ,即  $m$  的值仍为 6。第二次外循环  $i$  的值为 2, $j$  的值为 3,满足条件,执行 break 语句,跳出循环。

(23)A) 【解析】本题中第一次调用为  $\text{fun}(8, \text{fun}(5,6))$ ,因为  $\text{fun}(5,6)$  返回值为 5,所以第二次调用为  $\text{fun}(8,5)=6$ 。所以选择答案 A)。

(24)D) 【解析】D)选项中  $*f(a,b)$  表示调用后返回一个指向整型数据的地址指针,即该函数的返回值为指针类型,所以不能将其赋值给整型变量  $k$ 。

(25)A) 【解析】选项 B)的正确写法应为  $p = \&n$ ;选项 C)的正确写法应为  $\text{scanf}("%d", p)$ ;选项 D)的正确写法应为  $\text{printf}("%d\n", *p)$ 。

(26)A) 【解析】数组说明的一般形式为:类型说明符 数组名 [常量表达式]。B)选项中  $N$  是变量,不能用变量定义数组长度。C)选项中数组长度是非法的一串数字。定义数组时必须为其指明长度,D)选项中数组长度为空,所以非法。

(27)D) 【解析】第一次循环  $i$  为 0, $i\%2$  为 0,执行  $\text{switch}(a[0]\%2)$  中的 case 0 语句后内容即  $a[0]++$ , $a[0]$  的值变成 3;第二次循环  $i$  的值为 1, $i\%2$  为 1,执行 case 1:  $a[i]=0$ ;所以  $a[1]$  的值变成 0;第三次循环  $i$  的值为 2, $i\%2$  为 0,执行  $\text{switch}(a[2]\%2)$  中的 case 1 语句后的内容  $a[2]--$ , $a[2]$  的值变成 4;第四次循环  $i$  的值为 3, $i\%2$  为 1,执行 case 1:  $a[i]=0$ ;所以  $a[3]$  的值变成 0。

(28)B) 【解析】每次内循环只循环一次就结束,第一次外循环时  $t=t+b[0][b[0][0]]=1+b[0][0]=1+0=1$ ;第二次外循环时  $t=t+b[1][b[1][1]]=1+b[1][1]=1+1=2$ ;第三次外循环时  $t=t+b[2][b[2][2]]$

$=2+b[2][2]=2+2=4$ 。

(29)A) 【解析】数组定义后,不可以对数组整体赋值,ss 是二维数组,因 ss[1] 是一维字符数组,即字符串,字符串赋值可以使用 strcpy(ss[1], "right"); 这样的形式,而选项 A) 中对二维数组中的第“1”维(相当于一个一维数组)赋值,是不可以的。选项 B) 和 D) 是定义时对数组初始化,这是可以的。选项 C) 中,将字符串在内存中的首地址赋给指针数组的一个元素,这是可以的。

(30)A) 【解析】do { \*t++ = \*s++; } while( \*s); 不能因为当 \*s = '\0' 时,while( \*s) 跳出循环,这样字符串结束标志 '\0' 没有复制给 \*t,造成 \*t 不完整。注意, \*t++ = \*s++ 是先执行 t = \*s, 然后才进行 t = t + 1, s = s + 1。B)、C)、D) 都能将 '\0' 复制过去。

(31)D) 【解析】字符串是一个特殊的数组,所以按照数组的规则,s1 应该指向的是数组的首地址,即“abc”的第一个字符的地址。s2 指向的是“123”的第一个字符的地址。调用 swap 函数之后交换的是两个字符串的第一个字符‘a’和‘1’的内容,所以打印输出为 D)。

(32)C) 【解析】本题中由循环条件可知遇到 '\0' 或 x 与 y 所指的字符的值不等中的一个条件时就结束,所以功能是统计 x 和 y 所指字符串中最前面连续相同的字符个数。

(33)D) 【解析】本题目中静态局部变量 x,在静态存储区内分配存储单元。在程序整个运行期间都不释放。所以第一次循环 s 的值为 2;第二次循环中,返回的 x 的值为 4,所以 s 的值为 8;第三次循环中,返回的 x 的值为 8,所以 s 的值为 64。

(34)A) 【解析】全局变量的作用域是从声明处到文件的结束。所以选择 A)。

(35)A) 【解析】关键字 typedef 的作用只是将 C 语言中的已有的数据类型作了置换,并不是增加新的类型,所以 A) 错误。

(36)D) 【解析】声明 data 是结构 S 数组。初始化 data[0].a = 10; , data[0].b = 100; , data[1].a = 20; , data[1].b = 200。主函数中 p = data[1]; , 即 p.a = data[1].a; , p.b = data[1].b;。执行语句 printf("%d\n", ++(p.a)); 打印输出时 p.a 先增 1 再打印,即 p.a = data[1].a = 20, 先增 1 等于 21。

(37)C) 【解析】宏定义不是 C 语句,末尾不需要有分号。所以语句 printf("%4.1f\n", S(a+b)); 展开后为 printf("%4.1f\n", 3.5; \*a+b\*a+b;); 所以程序会出现语法错误。

(38)B) 【解析】& 按位与,如果两个相应的二进制位都为 1,则该位的结果值为 1,否则为 0。| 按位或,两个相应的二进制位中只要有一个为 1,该位的结果值为 1。2 的二进制为 00000010,4 的二进制为 00000100,所以做或运算结果为 00000110,该数与 5 即 00000101 做与操作结果为 00000100,即 4。

(39)D) 【解析】malloc 函数动态分配一个整型的内存空间,然后把函数返回的地址用 (int \*) 强制类型转换为整型指针,再把它赋给 a,b,c,即让指针变量 a,b,c 都指向刚申请的内存空间。所以只有最后一个赋值语句 \*c = 3 的值保留在了该空间内,因为 a,b,c 3 个指针变量均指向该空间,所以打印该空间内的数值为 3。

(40)A) 【解析】B) 选项中打开一个已存在的文件并进行了写操作后,原有文件中的全部数据不一定被覆盖,也可以对源文件进行追加操作等。C) 选项中在一个程序中对文件进行了写操作后,不用先关闭该文件然后再打开就能读到第 1 个数据,可以用 fseek() 函数进行重新定位即可。D) 选项中,C 语言中的文件可以进行随机读写。

## 二、程序填空题

### 【参考答案】

(1) 0      (2) n      (3) (t \* t)

### 【考点分析】

本题考查:变量初始化操作,需注意该变量是用做累加器还是累乘器;for 循环语句。

### 【解题思路】

填空 1:程序开始定义了变量 s,但没有对其进行初始化,根据公式及后面的程序可知变量 s 用来存放公式的前 n 项和,因此该变量应初始化为 0。

填空 2:通过 for 循环语句将表达式各项进行累加,结果存于变量 s 中,循环变量 i 的取值范围为 1~n。

填空 3:根据题目要求确定表达式通项。前面已定义  $t = 2.0 * i$ ,因此此空应填( $t * t$ )。

**【解题宝典】**

程序要求计算表达式的值时,一般都设定一个变量 s 并赋初值。如果 s 用来存放表达式的和,则赋初值为 0;

如果 s 用来存放表达式的积,则赋初值为 1。

**三、程序修改题**

**【参考答案】**

(1)for(i=0;str[i];i++)

(2)if(substr[k+1]== '\0')

**【考点分析】**

本题考查:for 循环语句和 if 条件语句的格式。for 循环语句使用最为灵活,其一般形式为 for( 表达式 1;表达式 2;表达式 3),注意表达式之间使用“;”相隔;在 if 条件语句中,if 关键字需要区别大小写,不能混淆使用。关键字是由 C 语言规定的具有特定意义的字符串,也称为保留字。用户定义的标识符不应与关键字相同,并且关键字应小写。

**【解题思路】**

先看循环条件 for(i = 0, str[i], i++) ,不难发现此处 for 循环语句的格式有误,其中表达式之间应以“;”相隔;同时很容易发现 if 条件语句处的关键字书写错误。

**四、程序设计题**

**【参考答案】**

```
double fun(double eps)
{
 double s=1.0,s1=1.0;
 int n=1;
 while(s1>=eps) /* 当某项大于精度要求时,继续求下一项 */
 {
 s1=s1*n/(2*n+1); /* 求多项式的每一项 */
 s=s+s1; /* 求和 */
 n++;
 }
 return 2*s;
}
```

**【考点分析】**

本题考查:迭代法求给定多项式的值。迭代算法是让计算机对一组指令(或一定步骤)进行重复执行,在每次执行这组指令(或这些步骤)时,都从变量的原值推出它的一个新值。需要注意变量的数据类型以及赋初值操作。

**【解题思路】**

首先应该定义 double 类型变量,并且赋初值,用来存放多项式的某一项和最后的总和。从第二项开始以后的每一项都是其前面一项乘以  $n/(2 * n + 1)$ ,程序中用 s1 来表示每一项,s 表示求和后的结果。需注意 s1 和 s 的初值都为 1.0,因为循环变量从第二项开始累加。

2013 年 3 月全国计算机等级考试无纸化二级 C 语言  
无纸化题库试题

(考试时间 120 分钟,满分 100 分)

一、选择题(每小题 1 分,共 40 分)

- (1) 程序流程图中带有箭头的线段表示的是( )。  
A)图元关系                      B)数据流                      C)控制流                      D)调用关系
- (2) 结构化程序设计的基本原则不包括( )。  
A)多态性                      B)自顶向下                      C)模块化                      D)逐步求精
- (3) 软件设计中模块划分应遵循的准则是( )。  
A)低内聚低耦合                      B)高内聚低耦合                      C)低内聚高耦合                      D)高内聚高耦合
- (4) 在软件开发中,需求分析阶段产生的主要文档是( )。  
A)可行性分析报告                      B)软件需求规格说明书  
C)概要设计说明书                      D)集成测试计划
- (5) 算法的有穷性是指( )。  
A)算法程序的运行时间是有限的                      B)算法程序所处理的数据量是有限的  
C)算法程序的长度是有限的                      D)算法只能被有限的用户使用
- (6) 对长度为  $n$  的线性表排序,在最坏情况下,比较次数不是  $n(n-1)/2$  的排序方法是( )。  
A)快速排序                      B)冒泡排序                      C)直接插入排序                      D)堆排序
- (7) 下列关于栈的叙述正确的是( )。  
A)栈按“先进先出”组织数据                      B)栈按“先进后出”组织数据  
C)只能在栈底插入数据                      D)不能删除数据
- (8) 在数据库设计中,将 E-R 图转换成关系数据模型的过程属于( )。  
A)需求分析阶段                      B)概念设计阶段                      C)逻辑设计阶段                      D)物理设计阶段
- (9) 有三个关系 R、S 和 T 如下:

| R |   |    | S |   |    | T |   |    |
|---|---|----|---|---|----|---|---|----|
| B | C | D  | B | C | D  | B | C | D  |
| a | 0 | k1 | f | 3 | h2 | a | 0 | k1 |
| b | 1 | n1 | a | 0 | k1 |   |   |    |
|   |   |    | n | 2 | x1 |   |   |    |

- 由关系 R 和 S 通过运算得到关系 T,则所使用的运算为( )。  
A)并                      B)自然连接                      C)笛卡儿积                      D)交
- (10) 设有表示学生选课的一张表,学生 S(学号,姓名,性别,年龄,身份证号),课程 C(课号,课名),选课 SC(学号,课号,成绩),则表 SC 的关键字(键或码)为( )。  
A)课号,成绩                      B)学号,成绩                      C)学号,课号                      D)学号,姓名,成绩
- (11) 以下叙述中错误的是( )。  
A)C 语言中的每条可执行语句和非执行语句最终都将被转换成二进制的机器指令  
B)C 程序经过编译、连接步骤之后才能形成一个真正可执行的二进制机器指令文件  
C)用 C 语言编写的程序称为源程序,它以 ASCII 代码形式存放在一个文本文件中  
D)C 语言源程序经编译后生成后缀为 .obj 的目标程序



```
}
```

程序输出( )。

A)1,1,2,0

B)0,0,0,3

C)编译有错

D)0,1,2,0

(20)有以下程序:

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

```
main()
```

```
{ int x = 1, y = 0, a = 0, b = 0;
```

```
switch (x)
```

```
{ case 1:
```

```
switch(y)
```

```
{ case 0: a ++; break;
```

```
case 1: b ++; break;
```

```
}
```

```
case 2: a ++; b ++; break;
```

```
case 3: a ++; b ++;
```

```
}
```

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

```
}
```

程序的运行结果是( )。

A)a=2,b=2

B)a=2,b=1

C)a=1,b=1

D)a=1,b=0

(21)以下程序段中的变量已正确定义:

```
for(i=0; i<4; i ++,i ++)
```

```
for(k=1; k<3; k ++); printf(" * ");
```

程序段的输出结果是( )。

A) \* \*

B) \* \* \* \*

C) \*

D) \* \* \* \* \* \* \* \*

(22)有以下程序段:

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

```
int i, n;
```

```
for(i=0; i<8; i ++)
```

```
{ n = rand() % 5;
```

```
switch (n)
```

```
{ case 1:
```

```
case 3: printf("%d\n", n); break;
```

```
case 2:
```

```
case 4: printf("%d\n", n); continue;
```

```
case 0: exit(0);
```

```
}
```

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

```
}
```

以下关于程序段执行情况的叙述,正确的是( )。

A)for 循环语句固定执行 8 次

B)当产生的随机数 n 为 4 时结束循环操作

C)当产生的随机数 n 为 1 和 2 时不做任何操作

D)当产生的随机数 n 为 0 时结束程序运行

(23)有以下程序:

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

```
int f(int x);
```

```
main()
```

```
{ int n = 1, m;
```



```
m=f(f(f(n))); printf("%d\n",m);
```

```
}
```

```
int f(int x)
```

```
{ return x * 2; }
```

程序运行后的输出结果是( )。

A)8

B)2

C)4

D)1

(24)以下叙述中错误的是( )。

A)可以给指针变量赋一个整数作为地址值

B)函数可以返回地址值

C)改变函数形参的值,不会改变对应实参的值

D)当在程序的开头包含头文件 stdio. h 时,可以给指针变量赋 NULL

(25)设已有定义:float x;,则以下对指针变量 p 进行定义且赋初值的语句中正确的是( )。

A)int \*p=(float)x;

B)float \*p=&x;

C)float p=&x;

D)float \*p=1024;

(26)以下数组定义中错误的是( )。

A)int x[2][3]={1,2,3,4,5,6};

B)int x[][3]={0};

C)int x[][3]={1,2,3},{4,5,6};

D)int x[2][3]={1,2},{3,4},{5,6};

(27)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");
```

```
}
```

程序的运行结果是( )。

A)321678

B)876543

C)1098765

D)345678

(28)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] - a[i][N-1-i];
```

```
}
```

```
main()
```

```
{ int x[N][N]={ {1, 2, 3, 4}, {5, 6, 7, 8}, {9,10,11,12}, {13,14,15,16}}, y[N], i;
```

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

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

```
}
```

程序运行后的输出结果是( )。

A) -3, -1,1,3,

B) -12, -3,0,0,

C)0,1,2,3,

D) -3, -3, -3, -3,

(29)设有定义:char s[81]; int i=0;,以下不能将一行(不超过80个字符)带有空格的字符串正确读入的语句或语句组是( )。

A)gets(s);

- B) while((s[i++] = getchar()) != '\n'); s[i] = '\0';  
C) scanf("%s", s);  
D) do{ scanf("%c", &s[i]); } while(s[i++] != '\n'); s[i] = '\0';
- (30) 设有定义: char p[] = {'1', '2', '3'}, \*q = p; , 以下不能计算出一个 char 型数据所占字节数的表达式是( )。
- A) sizeof(p)                      B) sizeof(char)                      C) sizeof(\*q)                      D) sizeof(p[0])

(31) 有以下程序:

```
#include <stdio.h>
#include <string.h>
main()
{
 char str[][20] = {"One * World", "One * Dream!"}, *p = str[1];
 printf("%d", strlen(p)); printf("%s\n", p);
}
```

程序运行后的输出结果是( )。

- A) 10, One \* Dream!    B) 9, One \* Dream!                      C) 9, One \* World                      D) 10, One \* World
- (32) 有以下程序:

```
#include <stdio.h>
main()
{
 char s[] = "012xy\08s34f4w2";
 int i, n = 0;
 for (i = 0; s[i] != '\0'; i++)
 if(s[i] >= '0' && s[i] <= '9') n++;
 printf("%d\n", n);
}
```

程序运行后的输出结果是( )。

- A) 0                      B) 3                      C) 7                      D) 8
- (33) 有以下程序:

```
#include <stdio.h>
fun(int x, int y)
{
 static int m = 0, i = 2;
 i++ = m + 1; m = i + x + y; return m;
}
main()
{
 int j = 1, m = 1, k;
 k = fun(j, m); printf("%d", k);
 k = fun(j, m); printf("%d\n", k);
}
```

执行后的输出结果是( )。

- A) 5, 11                      B) 5, 5                      C) 11, 11                      D) 11, 5
- (34) 在 C 语言中, 只有在使用时才占用内存单元的变量, 其存储类型是( )。
- A) auto 和 static                      B) extern 和 register                      C) auto 和 register                      D) static 和 register

(35) 下面结构体的定义语句中, 错误的是( )。

- A) struct ord {int x; int y; int z;} struct ord a;                      B) struct ord {int x; int y; int z;} ; struct ord a;  
C) struct ord {int x; int y; int z;} a;                      D) struct {int x; int y; int z;} a;

(36) 有以下程序:

```
#include <stdio.h>
typedef struct { int b, p; } A;
```

void f(A c) /\* 注意:c 是结构变量名 \*/

```
{ int j;
 c.b += 1; c.p += 2;
}
main()
{ int i;
 A a = {1,2};
 f(a);
 printf(" %d,%d\n", a.b, a.p);
}
```

程序运行后的输出结果是( )。

- A)2,4                      B)1,2                      C)1,4                      D)2,3

(37)以下叙述中正确的是( )。

- A)在 C 语言中,预处理命令行都以“#”开头  
B)预处理命令行必须位于 C 源程序的起始位置  
C)#include <stdio.h> 必须放在 C 程序的开头  
D)C 语言的预处理不能实现宏定义和条件编译的功能

(38)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));
}
```

程序的运行结果是( )。

- A)0                      B)10                      C)20                      D)出错

(39)有以下程序:

```
include <stdio.h>
main()
{ unsigned char a=8, c;
 c = a > 3;
 printf(" %d\n", c);
}
```

程序运行后的输出结果是( )。

- A)16                      B)1                      C)32                      D)0

(40)读取二进制文件的函数调用形式为:fread( buffer,size,count,fp);,其中 buffer 代表的是( )。

- A)一个内存块的字节数  
B)一个整型变量,代表待读取的数据的字节数  
C)一个文件指针,指向待读取的文件  
D)一个内存块的首地址,代表读入数据存放的地址

二、程序填空题(共 18 分)

下列给定程序中,函数 fun 的功能是:计算如下公式

$$S = \frac{3}{2^2} - \frac{5}{4^2} + \frac{7}{6^2} - \dots + (-1)^{n-1} \frac{(2 \times n + 1)}{(2 \times n)^2}$$

直到  $\left| \frac{(2 \times n + 1)}{(2 \times n)^2} \right| \leq 10^{-3}$ , 并且把计算结果作为函数值返回。

例如,若形参 e 的值为  $1e-3$ ,则函数返回值为 0.551690。

请在下画线处填入正确的内容并将下画线删除,使程序得出正确的结果。

注意:部分源程序给出如下。

不得增行或删行,也不得更改程序的结构!

试题程序:

```
#include <stdio.h>

double fun(double e)
{
 int i, k; double s, t, x;
 s=0; k=1; i=2;
 /***** found*****/
 x=1/4;
 /***** found*****/
 while(x2e)
 {
 s=s+k*x;
 k=k*(-1);
 t=2*i;
 /***** found*****/
 x=3/(t*t);
 i++;
 }
 return s;
}

main()
{
 double e=1e-3;
 printf("\nThe result is: %f\n",fun(e));
}
```

### 三、程序修改题(共 18 分)

下列给定程序中函数 fun 的功能是:求出如下分数序列的前 n 项之和,和值通过函数值返回。

$$\frac{2}{1}, \frac{3}{2}, \frac{5}{3}, \frac{8}{5}, \frac{13}{8}, \frac{21}{13}, \dots$$

例如,若  $n=5$ ,则应输出 8.391667。

请改正程序中的错误,使其得出正确的结果。

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!

试题程序:

```
#include <stdlib.h>
#include <conio.h>
#include <stdio.h>
/***** found*****/
void fun (int n)
{
 int a=2, b=1, c, k;
 double s=0.0;
 for(k=1;k<=n;k++)
 {
```

```

/***** found*****/
 s=s+(Double)a/b;
 c=a;a=a+b;b=c;
}
return(s);
}
void main()
{
 int n=5;
 system("CLS");
 printf("\nThe value of function is:%1f\n", fun(n));
}

```

#### 四、程序设计题(共 24 分)

请编写函数 fun,其功能是:找出  $2 \times M$  整型二维数组中最大元素的值,并将此值返回调用函数。

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 的花括号中填入你编写的若干语句。

试题程序:

```

#define M 4
#include <stdio.h>
int fun (int a[][M])
{

}

void main()
{
 int arr[2][M]={5, 8, 3, 45, 76, -4, 12, 82};
 printf("max=%d\n", fun(arr));
}

```

## 无纸化题库试题答案解析

### 一、选择题

(1)C) 【解析】在数据流图中,用标有名字的箭头表示数据流。在程序流程图中,用标有名字的箭头表示控制流。所以选择 C)。

(2)A) 【解析】结构化程序设计的思想包括:自顶向下、逐步求精、模块化、限制使用 goto 语句,所以选择 A)。

(3)B) 【解析】软件设计中模块划分应遵循的准则是高内聚低偶合、模块大小规模适当、模块的依赖关系适当等。模块的划分应遵循一定的要求,以保证模块划分合理,并进一步保证以此为依据开发出的软件系统可靠性强,易于理解和维护。模块之间的耦合应尽可能的低,模块的内聚度应尽可能的高。

(4)B) 【解析】A)错误,可行性分析阶段产生可行性分析报告。C)错误,概要设计说明书是总体设计阶段产生的文档。D)错误,集成测试计划是在概要设计阶段编写的文档。B)正确,需求规格说明书是后续工作如设计、编码等需要的重要参考文档。

(5)A) 【解析】算法原则上能够精确地运行,而且人们用笔和纸做有限次运算后即可完成。有穷性是指算法程序的运行时间是有限的。

(6)D) 【解析】除了堆排序算法的比较次数是  $O(n \log_2 n)$ ,其他的都是  $n(n-1)/2$ 。

(7)B) 【解析】栈是按“先进后出”的原则组织数据的,数据的插入和删除都在栈顶进行操作。

(8)C) 【解析】将 E-R 图转换成关系模型数据是把图形分析出来的联系反映到数据库中,即设计出表,所

以属于逻辑设计阶段。

(9)D) 【解析】自然连接是一种特殊的等值连接,它要求两个关系中进行比较的分量必须是相同的属性组,并且在结果中把重复的属性列去掉,所以 B) 错误。笛卡儿积是用 R 集合中元素为第一元素,S 集合中元素为第二元素构成的有序对,所以 C) 错误。根据关系 T 可以很明显地看出其是从关系 R 与关系 S 中取得相同的关系组成,所以运用的是交运算,选择 D)。

(10)C) 【解析】学号是学生表 S 的主键,课号是课程表 C 的主键,所以选课表 SC 的关键字就应该是与前两个表能够直接联系且能唯一定义的学号和课号,所以选择 C)。

(11)A) 【解析】C 语言中的非执行语句不会被编译,不会生成二进制的机器指令,所以 A) 错误。由 C 语言构成的指令序列称为 C 源程序,C 源程序经过 C 语言编译程序编译之后生成一个后缀为 .OBJ 的二进制文件(称为目标文件);最后要由“连接程序”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后缀为 .EXE 的可执行文件。

(12)A) 【解析】C 语言中八进制整型常量的开头是数字 0,十六进制整型常量的开头数字是 0x。C 语言中的实型常量有两种表示形式:小数形式,小数形式表示的实型常量必须要有小数点;指数形式,以“e”或“E”后跟一个整数来表示以 10 为底数的幂数,且规定字母 e 或 E 之前必须要有数字,且 e 或 E 后面的指数必须为整数。B) 选项中 028 错误,八进制中没有数字 8。C) 选项中 4e1.5 中 e 后面的指数不是整数。D) 选项中 3.e5 小数点后面缺少数字。

(13)A) 【解析】C 语言中标识符由字母、下划线、数字组成,且开头必须是字母或下划线。另外,关键字不能作为标识符。因为 C 语言中区分大小写,所以 B) 选项中的“FOR”可以作为标识符来用。A) 选项中含有非法字符 &,所以选择 A)。

(14)A) 【解析】B) 选项中函数 pow(x,e) 错误,应该直接使用 exp(x) 函数。C) 选项中函数 abs(n^x + e^x) 错误,应该使用 fabs() 返回浮点数的绝对值。D) 选项中 pow(x,n) 参数顺序错误。

(15)B) 【解析】取模运算符“%”为二元运算符,具有左结合性,并且参与运算的量应均为整型。选项 B 中的 a 变量是 double 实型,所以 B) 不符合规定。

(16)D) 【解析】C 语言中利用 sizeof() 函数判断数据类型长度,在 VC 6.0 平台中,整型 int 占有 4 个字节,double 型数据占有 8 个字节。

(17)D) 【解析】按照从键盘输入的数据可以判断字符 1 给了变量 a,字符 2 给了变量 b,字符 <CR> 即回车给了变量 c,字符 3 给了变量 d。所以打印输出的结果为 D) 选项。

(18)A) 【解析】C 语言的逻辑运算符比较特别,它的操作数没有明确的数据类型,可以是任意合法的表达式,所以选择 A)。

(19)C) 【解析】本题中 if(a=1)b=1; 与 else d=3; 之间多了语句 c=2;, 所以会出现 else 语句的位置错误的编译失败提示。

(20)B) 【解析】case 常量表达式只是起语句标号作用,并不是该处进行条件判断。在执行 switch 语句时,根据 switch 的表达式,找到与之匹配的 case 语句,从此 case 子句执行下去,不再进行判断,直到碰到 break 或函数结束为止。所以执行内层 switch(y) 时只执行了 a++, 此时 a 的值为 1。然后执行外层 case 2 语句的 a++;b++; 则 a 为 2,b 为 1。所以结果为 B)。

(21)C) 【解析】由于内层循环 for(k=1; k<3; k++) 后面直接跟了空语句“;”,所以在循环内部什么操作也不做,跳出外层循环后执行打印语句,所以打印了一个“\*”,因此选择 C)。

(22)D) 【解析】case 常量表达式只是起语句标号作用,并不是该处进行条件判断。在执行 switch 语句时,根据 switch 的表达式,找到与之匹配的 case 语句,从此 case 子句执行下去,不再进行判断,直到碰到 break 或函数结束为止。简单地说,break 是结束整个循环体,而 continue 是结束单次循环。B) 选项中当产生的随机数 n 为 4 时,要执行打印操作。C) 选项中当产生的随机数为 1 和 2 时,分别执行 case 3 与 case 4 后面语句的内容。由于存在 break 语句,所以 for 循环不是固定执行 8 次,执行次数与产生的随机数 n 有关系。

(23)A) 【解析】第一次调用为 m=f(f(f(1))), 第二次调用为 m=f(f(2)), 第三次调用为 m=f(4), 即返回值为 8。

(24)A) 【解析】不能将一个整数直接赋给指针变量作为地址,所以 A) 是错误的。函数的返回值可以是地址,即指针。函数调用中形参值的变化不会传递给实参。

(25)B) 【解析】指针是用来存放地址的变量,用(类型名 \* 指针变量名)的形式定义。赋值时应将某个变

量地址即 &x 赋给指针变量,所以选择 B)。

(26)D) 【解析】D) 选项中 `x[2][3]` 定义的是一个两行三列的二维数组,而在给数组元素赋值时却赋成了三行,所以错误。

(27)A) 【解析】本题中的函数 `fun()` 的功能是将数组 `k` 中前 5 个元素倒序,所以返回后数组 `k` 中的元素排列是 5,4,3,2,1,6,7,8,9,10。所以打印输出 `k[2]` 到 `k[7]` 元素的值,即 321678,所以选择 A)。

(28)A) 【解析】本题中由 `fun` 函数可知, $b[0] = a[0][0] - a[0][3] = 1 - 4 = -3$ , $b[1] = a[1][1] - [1][2] = 6 - 7 = -1$ , $b[2] = a[2][2] - [2][1] = 11 - 10 = 1$ , $b[3] = a[3][3] - [3][1] = 16 - 13 = 3$ ,所以主函数中打印 `y` 数组元素的值为 A) 选项。

(29)C) 【解析】字符串的输入不能使用 `scanf("%s",s);`,而应该使用 `gets(s)`。

(30)A) 【解析】根据题目中的定义,可以知道 `sizeof(p)` 计算的是数组 `p` 中所有元素所占用的字节数,而不是 `char` 型数据所占字节数。

(31)A) 【解析】`p` 是指向二维字符数组第二行 `One * Dream!` 的数组指针,所以长度是 10,打印输出的也是该字符串。

(32)B) 【解析】本题中遇到“\”字符循环结束,所以只统计“\”之前的数字字符,所以为 3。

(33)A) 【解析】声明静态局部变量:函数调用结束后,其占用的存储单元不释放,在下次该函数调用时,该变量保留上一次函数调用结束时的值。本题子函数 `fun` 中的变量 `i` 和 `m` 均为静态局部变量。所以第一次调用 `fun` 函数,返回 `m` 的值为 5,第二次再调用 `fun` 函数时,`i` 的值为 3,`m` 的值已经是 5 了,所以执行 `i + = m + 1`,`i` 的值变为 9, $m = i + x + y = 9 + 1 + 1 = 11$ 。

(34)C) 【解析】`auto`:函数中的局部变量,动态地分配存储空间,数据存储在动态存储区中,在调用该函数时系统会给它们分配存储空间,在函数调用结束时就自动释放这些存储空间。`register`:为了提高效率,C 语言允许将局部变量的值放在 CPU 中的寄存器中,这种变量称为寄存器变量,只有局部自动变量和形式参数可以作为寄存器变量。`extern`:外部变量(即全局变量)是在函数的外部定义的,它的作用域为从变量定义处开始,到本程序文件的末尾。如果外部变量不在文件的开头定义,其有效的作用范围只限于定义处到文件终了。`static`:静态局部变量,属于静态存储类别,在静态存储区内分配存储单元,在程序整个运行期间都不释放。

(35)A) 【解析】A) 选项 `struct ord {int x;int y;int z;} struct ord a;` 错误,不能在定义结构体的同时,又用结构体类型名定义变量。应该写成 B) 选项或者 D) 选项的格式。

(36)B) 【解析】结构体变量可以作为函数的参数和返回值。作为函数的实参时,可以实现函数的传值调用。当使用结构体变量作为函数的形参时,实参也应该是结构体变量名,以实现传值调用,实参将复制副本给形参,在被调用函数中改变形参值对调用函数中的实参没有影响。所以选择 B)。

(37)A) 【解析】预处理命令是以“#”号开头的命令,它们不是 C 语言的可执行命令,这些命令应该在函数之外书写,一般在源文件的最前面书写,但不是必须在起始位置书写,所以 B)、C) 错误。C) 语言的预处理能够实现宏定义和条件编译等功能,所以 D) 错误。

(38)C) 【解析】`malloc(sizeof(int))` 的作用是开辟一个长度为 `sizeof(int)` 存储空间,并通过强制类型转换(`int *`)将此存储空间的地址赋给了一个整型的指针变量 `p`。然后执行语句 `*p = n`,使得 `*p` 的值为 10,并通过 `return` 返回此值,即 `a` 的值为 10。然后在主函数中输出 `a + fun(10) = 10 + 10 = 20`。

(39)B) 【解析】无符号整型变量 `a` 的值为 8,二进制表示为 00001000,右移 3 位后为 00000001,即十进制的 1,所以输出 1。

(40)D) 【解析】`fread(void *buffer, size_t size, size_t count, FILE *stream)`;的功能是从一个文件流中读取数据,读取 `count` 个元素,每个元素 `size` 字节,如果调用成功返回 `count`。`buffer`:用于接收数据的内存地址,大小至少是 `size * count` 字节;`size`:单个元素的大小,单位是字节;`count`:元素的个数,每个元素是 `size` 字节;`stream`:输入流。

## 二、程序填空题

### 【参考答案】

(1) 3.0 或 (double)3      (2) >      (3) (t+1)

### 【考点分析】

本题考查:基本算术运算的类型转换。其转换规则如下:

如果双目运算符两边运算数的类型一致,则所得结果的类型与运算数的类型一致。

如果双目运算符两边运算数的类型不一致,系统将自动进行类型转换,使运算符两边运算数的类型达到一致

后再进行运算。其类型转换规律见下表。

| 运算数 1 | 运算数 2 | 转换结果类型      |
|-------|-------|-------------|
| 短整型   | 长整型   | 短整型→长整型     |
| 整型    | 长整型   | 整型→长整型      |
| 字符型   | 整型    | 字符型→整型      |
| 有符号整型 | 无符号整型 | 有符号整型→无符号整型 |
| 整型    | 浮点型   | 整型→浮点型      |

while 循环语句,一般考查如何根据题意确定循环条件,以及如何通过循环变量的变化实现循环操作。

【解题思路】

填空 1:变量 x 定义为 double 类型,而运算符“/”后面是整型数,因此给 x 赋值时,需要注意数据类型,此处不能将3.0写成3。

填空 2:while 循环语句的循环条件。根据题意确定循环变量应大于  $1e-3$ ,因此此处应填“>”。

填空 3:表达式的通项是  $(2 * i + 1) / (2 * i)^2$ ,由于程序中已定义  $t = 2 * i$ ,所以此处应该填写  $(t + 1)$ 。

【解题宝典】

本题填空 1 还可以运用强制类型转换来实现。强制类型转换是通过类型转换运算来实现的,其一般形式为“(类型说明符)(表达式)”。其功能是:把表达式的运算结果强制转换成类型说明符所表示的类型。

例如:

- (float) a            把 a 转换为实型
- (int)(x + y)        把 x + y 的结果转换为整型

三、程序修改题

【参考答案】

- (1)double fun(int n)
- (2)s = s + (double)a/b;

【考点分析】

本题考查:函数定义,函数定义的一般形式为“类型标识符函数名(形式参数表列)”,其中类型标识符用来说明函数值的类型,函数的类型实际上是函数返回值的类型。

【解题思路】

- (1)由语句“return(s);”和变量 s 定义为 double 型可知,该函数定义时其类型标识符为 double 类型。
- (2)注意数据类型 double 的书写格式。

【解题宝典】

确定函数类型标识符在填空和改错题中经常考到,如果函数没有返回值,则函数定义为 void 型;如果函数有返回值,则函数类型应与返回值类型一致;如果不定义函数类型标识符,则系统默认为 int 型。

四、程序设计题

【参考答案】

```
int fun (int a[][M])
{
 int i,j,max=a[0][0];
 /* 对二维数组进行遍历 */
 for(i=0;i<2;i++)
 for(j=0;j<M;j++)
 if(max<a[i][j])
 /* 将最大值保存在 max 中 */
 max=a[i][j];
 return max;
}
```



### 【考点分析】

本题考查:求数组的最大值,需运用循环语句。因为数组是二维数值,所以应使用二层加 for 循环嵌套。使用 for 循环语句时需注意循环变量的取值范围。

### 【解题思路】

此类求最大值或最小值的问题,可以采用逐个比较的方式,对数组中所有元素遍历一遍,从中找出数组最大值或最小值。首先定义变量 max 用来存放数组的第一个元素的值,然后利用 for 循环逐个找出数组中的元素,并与 max 比较,如果元素值大于 max,则将该值赋予 max,循环结束后 max 的值即为数组最大值,最后将该值返回。

### 【解题宝典】

该类题目考查较多,需要掌握逐个比较的方法。对于  $m * n$  二维数组,如果采用逐个查找方法,代码实现如下:

```
for (i = 0; i < m; i++)
 for (j = 0; j < n; j++)
 if (a[i, j] > max)
 max = a[i, j]
```

## 笔试真题

一、选择题((1)~(10)、(21)~(40)每题2分,(11)~(20)每题1分,共70分)

(1)下列链表中,其逻辑结构属于非线性结构的是( )。

- (2) 设循环队列的存储空间为  $Q(1:35)$ , 初始状态为  $\text{front} = \text{rear} = 35$ 。现经过一系列入队与退队运算后,  $\text{front} = 15$ ,  $\text{rear} = 15$ , 则循环队列中的元素个数为( )。

- A) 16                      B) 20                      C) 0 或 35                      D) 15

(3) 下列关于栈的叙述中, 正确的是( )。

- A) 栈顶元素一定是最先入栈的元素  
B) 栈操作遵循先进后出的原则  
C) 栈底元素一定是最后入栈的元素  
D) 以上三种说法都不对

(4)在关系数据库中,用来表示实体间联系的是( )。

- A) 二维表                      B) 树状结构                      C) 属性                      D) 网状结构

(5) 公司中有多个部门和多名职员,每个职员只能属于一个部门,一个部门可以有多名职员。则实体部门和职员间的联系是( )。

- A) m:1 联系      B) 1:m 联系      C) 1:1 联系      D) m:n 联系

(6) 有两个关系 R 和 S 如下:

| R |   |   | S |   |   |
|---|---|---|---|---|---|
| A | B | C | A | B | C |
| a | 1 | 2 | c | 3 | 1 |
| b | 2 | 1 |   |   |   |
| c | 3 | 1 |   |   |   |

则由关系 R 得到关系 S 的操作是( )。

- A) 自然连接      B) 选择      C) 并      D) 投影

(7)数据字典(DD)所定义的对象都包含于( )。

- A) 程序流程图                      B) 数据流图(DFD图)  
C) 方框图                          D) 软件结构图

(8)软件需求规格说明书的作用不包括( )。

- A) 软件可行性研究的依据  
B) 用户与开发人员对软件要做什么的共同理解  
C) 软件验收的依据  
D) 软件设计的依据

(9)下面属于黑盒测试方法的是( )。

- A) 逻辑覆盖      B) 语句覆盖      C) 路径覆盖      D) 边界值分析



```

while(a ++ && ++b);
printf(" %d,%d\n",a,b);
}

```

程序运行后的输出结果是( )。

- A)1,3                      B)0,2                      C)0,3                      D)1,2

(21)设有定义: int x=0, \*p;,立刻执行以下语句,正确的语句是( )。

- A)p=x;                      B)\*p=x;  
C)p=NULL;                      D)\*p=NULL;

(22)下列叙述中正确的是( )。

- A)可以用关系运算符比较字符串的大小  
B)空字符串不占用内存,其内存空间大小是0  
C)两个连续的单引号是合法的字符常量  
D)两个连续的双引号是合法的字符串常量

(23)有以下程序

```

#include <stdio.h>
main()
{
 char a='H';
 a=(a>='A'&&a<='Z')?(a-'A'+ 'a'): a;
 printf("%c\n",a);
}

```

程序运行后的输出结果是( )。

- A) A                      B)a                      C)H                      D)h

(24)有以下程序

```

#include <stdio.h>
int f(int x);
main()
{
 int a,b=0;
 for(a=0;a<3;a++)
 {
 b==b+f(a); putchar('A'+b);
 }
 int f(int x)
 {
 return x*x+1;
 }
}

```

程序运行后的输出结果是( )。

- A) ABE                      B)BDI                      C) BCF                      D) BCD

(25)设有定义: int x[2][3];,则以下关于二维数组 x 的叙述错误的是( )。

- A) x[0]可看作是由3个整型元素组成的一维数组  
B) x[0]和 x[1]是数组名,分别代表不同的地址常量  
C) 数组 x 包含6个元素  
D) 可以用语句 x[0]=0;为数组所有元素赋初值0

(26)设变量 p 是指针变量,语句 p=NULL;是给指针变量赋 NULL 值,它等价于( )。

- A) p="";                      B)p='0';                      C)p=0;                      D)p="";

(27)有以下程序

```

#include <stdio.h>
main()
{
 int a[]={10,20,30,40}, *p=a,i;
 for(i=0; i<=3; i++) {a[i]= *p; p++;}
 printf("%d\n",a[2]);
}

```

程序运行后的输出结果是( )。

A) 30

B) 40

C) 10

D) 20

(28)有以下程序

```
#include <stdio.h>
#define N 3
void fun(int a[][N],int b[])
{ int i,j;
 for(i=0;i<N;i++)
 { b[i]=a[i][0];
 for(j=1;j<N;j++)
 if(b[i]<a[i][j]) b[i]=a[i][j];
 }
}

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

程序运行后的输出结果是( )。

A) 2,4,8,

B) 3,6,9,

C) 3,5,7,

D) 1,3,5,

(29)有以下程序 (strcpy 为字符串复制函数, strcat 为字符串连接函数)

```
#include <stdio.h>
#include <string.h>
main()
{ char a[10]="abc", b[10]="012", c[10]="xyz";
 strcpy(a+1,b+2);
 puts(strcat(a,c+1));
}
```

程序运行后的输出结果是( )。

A) a12xyz

B) 12xyz

C) a2yz

D) bc2yz

(30)以下选项中,合法的是( )。

A) char str3[] = { 'd', 'e', 'b', 'u', 'g', '\0' }

B) char str4; str4 = "hello world";

C) char name[10]; name = "china";

D) char str1[5] = "pass", str2[6]; str2 = str1;

(31)有以下程序

```
#include <stdio.h>
main()
{ char *s = "12134"; int k=0, a=0;
 while(s[k+1] != '\0')
 { k++;
 if(k%2==0) { a = a + (s[k] - '0' + 1); continue; }
 a = a + (s[k] - '0');
 }
}
```

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

```
}
```

程序运行后的输出结果是( )。

A)k=6 a=11

B)k=3 a=14

C)k=4 a=12

D)k=5 a=15

(32)有以下程序

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

```
main()
```

```
{ char a[5][10]= {"one", "two", "three", "four", "five"};
```

```
int i, j;
```

```
char t;
```

```
for(i=0; i<4; i++)
```

```
for(j=i+1; j<5; j++)
```

```
if(a[i][0]>a[j][0])
```

```
{ t=a[i][0]; a[i][0]=a[j][0]; a[j][0]=t; }
```

```
puts(a[1]);
```

```
}
```

程序运行后的输出结果是( )。

A) two

B) fix

C) two

D) owo

(33)有以下程序

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

```
int a=1, b=2;
```

```
void fun1(int a, int b)
```

```
{ printf("%d%d", a, b); }
```

```
void fun2()
```

```
{ a=3; b=4; }
```

```
main()
```

```
{ fun1(5,6); fun2();
```

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

```
}
```

程序运行后的输出结果是( )。

A) 1 2 5 6

B) 5 6 3 4

C) 5 6 1 2

D) 3 4 5 6

(34)有以下程序

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

```
void func(int n)
```

```
{ static int num=1;
```

```
num=num+n; printf("%d", num);
```

```
}
```

```
main()
```

```
{ func(3); func(4); printf("\n"); }
```

程序运行后的输出结果是( )。

A) 4 8

B) 3 4

C) 3 5

D) 4 5

(35)有以下程序

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

```
#include <stdlib.h>
```

```
void fun(int *p1, int *p2, int *s)
```

```
{ s=(int *)malloc(sizeof(int));
```

```

 *s = *p1 + *p2;
 free(s);
 }
main()
{ int a=1,b=40,*q=&a;
 fun(&a,&b,q);
 printf(" %d \n", *q);
}

```

程序运行后的输出结果是( )。

- A) 42                      B) 0                      C) 1                      D) 41

(36)有以下程序

```

#include <stdio.h>

struct STU{char name[9]; char sex; int score[2];};
void f(struct STU a[])
{ struct STU b={ "Zhao", 'm', 85, 90 };
 a[1]=b;
}

main()
{ struct STU c[2]={ { "Qian", 'f', 95, 92 }, { "Sun", 'm', 98, 99 } };
 f(c);
 printf(" %s, %c, %d, %d", c[0].name, c[0].sex, c[0].score[0], c[0].score[1]);
 printf(" %s, %c, %d, %d", c[1].name, c[1].sex, c[1].score[0], c[1].score[1]);
}

```

程序运行后的输出结果是( )。

- A) Zhao, m, 85, 90, Sun, m, 98, 99  
 B) Zhao, m, 85, 90, Qian, f, 95, 92  
 C) Qian, f, 95, 92, Sun, m, 98, 99  
 D) Qian, f, 95, 92, Zhao, m, 85, 90

(37)以下叙述中错误的是( )。

- A) 可以用 typedef 说明的新类型名来定义变量  
 B) typedef 说明的新类型名必须使用大写字母, 否则会出编译错误  
 C) 用 typedef 可以为基本数据类型说明一个新名称  
 D) 用 typedef 说明新类型的作用是用一个新的标识符来代表已存在的类型名

(38)以下叙述中错误的是( )。

- A) 函数的返回值类型不能是结构体类型, 只能是简单类型  
 B) 函数可以返回指向结构体变量的指针  
 C) 可以通过指向结构体变量的指针访问所指结构体变量的任何成员  
 D) 只要类型相同, 结构体变量之间可以整体赋值

(39)若有定义语句 int b=2; 则表达式 (b<<2)/(3||b) 的值是( )。

- A) 4                      B) 8                      C) 0                      D) 2

(40)有以下程序

```

#include <stdio.h>

main()
{ FILE *fp; int i, a[6]={1,2,3,4,5,6};
 fp=fopen("d2.dat", "w+");
 for(i=0; i<6; i++) fprintf(fp, " %d \n", a[i]);
 rewind(fp);
}

```

```

for(i=0;i<6;i++) fscanf(fp, "%d",&a[5-i]);
fclose(fp);
for(i=0;i<6;i++) printf("%d",a[i]);
}

```

程序运行后的输出结果是( )。

A)4,5,6,1,2,3,      B)1,2,3,3,2,1,      C)1,2,3,4,5,6,      D)6,5,4,3,2,1,

## 二、填空题(每空 2 分,共 30 分)

请将每一个空的正确答案写在答题卡【1】至【15】序号的横线上,答在试卷上不得分。

- (1)一棵二叉树共有 47 个结点,其中有 23 个度为 2 的结点。假设根结点在第 1 层,则该二叉树的深度为 【1】。
- (2)设栈的存储空间为 S(1:40),初始状态为 bottom=0,top=0。现经过一系列入栈与出栈运算后,top=20,则当前栈中有 【2】 个元素。
- (3)数据独立性分为逻辑独立性和物理独立性。当总体逻辑结构改变时,其局部逻辑结构可以不变,从而根据局部逻辑结构编写的应用程序不必修改,称为 【3】。
- (4)关系数据库中能实现的专门关系运算包括 【4】、连接和投影。
- (5)软件按功能通常可以分为应用软件、系统软件和支撑软件(或工具软件)。UNIX 操作系统属于 【5】 软件。
- (6)请写出与!(a<=b)等价的 C 语言表达式 【6】。
- (7)以下程序运行时从键盘输入:1.0 2.0,输出结果是:1.000000 2.000000,请填空。

```

#include <stdio.h>
main()
{
 double a; float b;
 scanf(" 【7】 ",&a,&b); printf("%f%f\n", a,b);
}

```

(8)有以下程序

```

#include <stdio.h>
main()
{
 int n1=0, n2=0, n3=0; char ch;
 while((ch=getchar())!='\n')
 switch(ch)
 {
 case'1': case'3': n1++;break;
 case'2': case'4': n2++;break;
 default: n3++;break;
 }
 printf("%d%d%d\n", n1,n2,n3);
}

```

若程序运行时输入 01234567! <回车>,则输出结果是 【8】。

(9)有以下程序

```

#include <stdio.h>
main()
{
 int i, sum=0;
 for(i=1; i<9; i+=2) sum+=i;
 printf("%d\n",sum);
}

```

程序运行后的输出结果是 【9】。

(10)有以下程序

```

#include <stdio.h>
main()
{
 int d, n=1234;
}

```



```

while(n !=0)
{
 d = n%10; n = n/10; printf("%d", d);
}

```

程序运行后的输出结果是 【10】。

(11)有以下程序

```

#include <stdio.h>
int k=7;
int *st(int *a)
{
 int *c=&k;
 if(*a>*c) c=a;
 return c;
}
main()
{
 int i=3, *p=&i, *r;
 r=st(p); printf("%d\n", *r);
}

```

程序运行后的输出结果是 【11】。

(12)以下程序的输出结果是 【12】。

```

#include <stdio.h>
#define N 3
#define M(n) (N+1)*n
main()
{
 int x;
 x=2*(N+M(2));
 printf("%d\n", x);
}

```

(13)若有定义语句:char str[]="0";, 则字符串 str 在内存中实际占 【13】 字节。

(14)有以下程序

```

#include <stdio.h>
int fun(int n)
{
 if(n==0) return(1);
 return(fun(n-1)*n);
}
main()
{
 int t;
 t=fun(3); printf("%d\n", t);
}

```

程序运行后的输出结果是 【14】。

(15)以下函数的功能是输出链表结点中的数据,形参指针 h 已指向如下链表,请填空。



```

struct slist{char data; struct slist *next;};
void fun(struct slist *h)
{
 s struct slist *p;
 p=h;
 while(p)
 { printf("%c ", p->data); p = 【15】; }
 printf("\n");
}

```

# 全国计算机等级考试二级 C 语言

## 上机真题

(考试时间 90 分钟,满分 100 分)

### 一、程序填空题

下列给定程序中已建立一个带头结点的单向链表,链表中的各结点按结点数据域中的数据递增有序链接。函数 fun 的功能是:把形参 x 的值放入一个新结点并插入链表中,使插入后各结点数据域中的数据仍保持递增有序。请在下画线处填入正确的内容并将下画线删除,使程序得出正确的结果。

注意:部分源程序给出如下。

不得增行或删行,也不得更改程序的结构!

试题程序:

```
#include <stdio.h>
#include <stdlib.h>
#define N 8
typedef struct list
{ int data;
 struct list * next;
} SLIST;
void fun(SLIST * h, int x)
{ SLIST * p, * q, * s;
 s = (SLIST *)malloc(sizeof(SLIST));
 /***** found*****/
 s->data = 1;
 q = h;
 p = h->next;
 while(p != NULL && x > p->data) {
 /***** found*****/
 q = 2;
 p = p->next;
 }
 s->next = p;
 /***** found*****/
 q->next = 3;
}
SLIST * creatlist(int * a)
{ SLIST * h, * p, * q; int i;
 h = p = (SLIST *)malloc(sizeof(SLIST));
 for(i = 0; i < N; i++)
 { q = (SLIST *)malloc(sizeof(SLIST));
 q->data = a[i]; p->next = q; p = q;
 }
 p->next = 0;
 return h;
}
```

```

}
void outlist(SLIST * h)
{
 SLIST * p;
 p=h->next;
 if (p==NULL)
 printf("\nThe list is NULL! \n");
 else
 {
 printf("\nHead");
 do { printf("->% d",p->data);
 p=p->next;
 } while (p!=NULL);
 printf("->End\n");
 }
}

main()
{
 SLIST * head; int x;
 int a[N]={11,12,15,18,19,22,25,29};
 head=creatlist(a);
 printf("\nThe list before inserting:\n");
 outlist(head);
 printf("\nEnter a number : ");
 scanf("% d",&x);
 fun(head,x);
 printf("\nThe list after inserting:\n");
 outlist(head);
}

```

## 二、程序改错题

下列给定程序中函数 fun 的功能是:计算正整数 num 各位上的数字之积。

例如,若输入 252,则输出应该是 20。若输入 202,则输出应该是 0。

请改正程序中的错误,使它能得出正确的结果。

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!

试题程序:

```

#include <stdio.h>
#include <conio.h>
long fun(long num)
{
 /***** found *****/
 long k;
 do
 {
 k*=num%10;
 /***** found *****/
 num \=10;
 }while (num);
 return (k);
}

main ()

```

```

{
 long n;
 printf("\n please enter a number:");
 scanf("% ld",&n);
 printf("\n% ld\n",fun(n));
}

```

### 三、程序设计题

编写函数 fun,其功能是:计算  $n$  门课程的平均分,结果作为函数值返回。

例如,若有 5 门课程的成绩是:90.5,72,80,61.5,55,则函数的值为 71.80。

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 的花括号中填入你编写的若干语句。

试题程序:

```

#include <stdio.h>
float fun (float * a, int n)
{

}

void main()
{
 float score[30]={90.5,72,80,61.5,55},aver;
 aver=fun(score,5);
 printf("\nAverage score is:% 5.2f\n",aver);
}

```

## 笔试真题答案解析

### 一、选择题

(1)D) 【解析】循环链表、双向链表、带链的栈都是线性结构,二叉链表是非线性结构二叉树的链式存储结构,只有它是非线性结构。此处答案为 D)。

(2)C) 【解析】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 指向队头元素的前一个位置。因此,从队头指针 front 指向的后一个位置到队尾指针 rear 指向的位置之间所有的元素均为队列中的元素。循环队列的初始状态为空,即  $rear = front = m$ ,每进行一次入队运算,队尾指针就进一。每进行一次出队运算,队头指针就进一。此题中  $rear = front = 15$ ,可能出现的情况是入队的元素全部出队,此时队列中元素个数为 0;也可能是执行入队出队的次数不一样,最终状态是队列为满的状态,此时队列中元素个数为 35。故答案为 C)。

(3)B) 【解析】栈(Stack)是限定在一端进行插入与删除的线性表。在栈中,允许插入与删除的这一端称为栈顶,而不允许插入与删除的另一端称为栈底。栈是按照“先进后出”或“后进先出”的原则组织数据的,因此,栈也称为“先进后出”表或“后进先出”表,所以答案为 B)。

(4)A) 【解析】在关系数据库中,实体与实体间的联系可以用关系(二维表)的形式来表示。本题答案为 A)。

(5)B) 【解析】两个实体间的联系可分为 3 种类型:①一对一联系,一对一的联系表现为主表中的一条记录与相关表中的一条记录相关联②一对多联系,一对多的联系表现为主表中的一条记录与相关表中的多条记录相关联③多对多联系,多对多的联系表现为主表中的多条记录与相关表中的多条记录相关联。在本题中一个部门可以有多位职员,每位职员只能属于一个部门。故答案为 B)。

(6)B) 【解析】选择运算又称为限制。它是指从一个关系(表)中找出满足一定条件的所有元组(记录),即在二维表中选取若干行。选择运算是根据某些条件对关系做水平分割,即选取符合条件的元组。从题目中所给关系可以看出由关系 R 得出关系 S 的操作是选择,故答案为 B)。

(7)B) 【解析】数据字典是指对数据的数据项、数据结构、数据流、数据存储、处理逻辑、外部实体等进行定

义和描述,其目的是对数据流图中的各个元素做出详细的说明。故答案为 B)。

(8)A) 【解析】软件需求规格说明书(Software Requirement Specification,SRS)是需求分析阶段的最终成果,是软件开发中的重要文档之一。软件需求规格说明书的作用包括:①便于用户、开发人员进行理解和交流。②反映出用户问题的结构,可以作为软件开发工作的基础和依据。③作为确认测试和验收的依据。

(9)D) 【解析】黑盒测试也称功能测试,它通过测试来检测每个功能是否都能正常使用。在测试中,把程序看做一个不能打开的黑盒子,在完全不考虑程序内部结构和内部特性的情况下,在程序接口进行测试,它只检查程序功能是否按照需求规格说明书的规定正常使用,程序是否能适当地接收输入数据而产生正确的输出信息。黑盒测试方法主要有等价类划分法、边界值分析法、错误推测法、因果法等,主要用于软件确认测试。故答案为 D)。

(10)D) 【解析】软件设计包括总体设计和详细设计,总体设计又包括最佳方案的设计、软件结构设计、数据结构及数据库设计;详细设计是总体设计的进一步的具体化,其基本任务有:为每个模块进行详细的算法设计;为模块内的数据结构进行设计;对数据库进行物理设计,即确定数据库的物理结构;界面设计;编写文档;评审。对详细设计成果进行审查和复审。故答案选 D)。

(11)B) 【解析】C 语言中,main 函数的位置可出现在任意位置,故 A)选项错误;C 语言中的任意源程序都可单独编译,故 C)选项错误;每个 C 语言程序可由若干源文件即后缀为 .c 的文件组成,而一个 C 程序只能有一个 main 函数,故 D)选项错误。

(12)A) 【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,且第一个字符必须为字母或下划线,排除选项 B);大写字母和小写字母被认为是两个不同的字符,如 A 和 a 是两个不同的标识符,故排除选项 C);C 语言中还规定标识符不能为 C 语言的关键字,故排除选项 D)。

(13)B) 【解析】A)选项中 3 个“9”以空格间隔,不能表示为一个常量;B)选项为合法的十六进制数值;C 语言中表示浮点数时,e 后面应出现整数,因此 C)、D)选项均错误。

(14)A) 【解析】C 语言是通过函数来实现模块化程序设计的。所以较大的 C 语言应用程序,往往是由多个函数组成的,每个函数分别对应各自的功能模块。

(15)C) 【解析】C 语言中,定义符号常量的格式为“#define 常量名 常量值”,故 C)选项错误。

(16)C) 【解析】C 语言规定:如果在“格式控制”字符串中除了格式说明以外还有其他字符,则在输入数据时应输入与这些字符相同的字符。下面输入是合法的:3,4 此时不用逗号而用空格或其他字符是不对的,故选 C)。

(17)C) 【解析】C 语言中,char 类型占 1 个字节。

(18)B) 【解析】A)选项中,3+4 结果为 7,故 7>6 的结果为真;B)选项中,3!=4 的结果为 1,进而判断 1>2 的结果为假;C)选项中,不管 1 左面结果如何,1 右面始终为真,故 C)选项的结果始终为真;D)选项中 3<4 结果为 1,故 1==1 的结果为真。

(19)D) 【解析】case 后面必须跟常量,故 A)选项出现语法错误;switch 后面的运算式必须出现在“( )”中,故 B)选项出现语法错误;C)选项中的 case 与数字间没有空格,故 C)选项出现语法错误。

(20)D) 【解析】此题考查自增符号++的使用。第一次对循环条件判断,b 先自增,此时 a=-2,b=1,满足循环,a 再自增 a=-1;第二次对循环条件判断,b 先自增,此时 a=-1,b=2,满足循环,a 再自增 a=0;第三次对循环条件判断,由于 a=0 不满足条件,所以 a 再自增 a=1,然后跳出循环,输出打印 a=1,b=2。

(21)C) 【解析】p 为指向 int 类型的指针,p 存放内存地址,故该指针类型不能赋值为常量值。A)选项为非正常的内存地址赋值,B)、D)选项类型不符。

(22)D) 【解析】C 语言中不支持关系运算符对字符串的操作,故 A)错误;空字符串只是值为空,所占用的内存大小不变,故 B)选项错误;C 语言存储的字符皆转换为整型存储,故 C)错。

(23)D) 【解析】在判断 a 的值‘H’是否在‘A’到‘Z’时返回结果为真,故执行(a-‘A’+‘a’),ASCII 中 A 十进制整数为 65,a 为 97,H 为 72,故 72-65+97=104 则为小 h 的十进制整数,故应选 D)。

(24)B) 【解析】首次执行时,a 值为 0,f(a)的值为 1,故 b 值为 1,‘A’值为 65,putchar(65+1),故输出‘B’,第二次执行时,a 值为 1,f(a)为 2,b 值为 3,putchar(65+3),故输出‘D’;第三次执行,a 为 2,f(a)为 5,故 b 值为 8,putchar(65+7),故输出‘I’。

(25)D) 【解析】int x[2][3]是一个 2 行 3 列的二维数组,可以看成是 x[0]和 x[1]两个一维数组组成,总共有 2\*3=6 个元素组成。x[0]和 x[1]分别代表的第一行和第二行的首地址常量。D)选项中的语句不能实现对整个二维数组的初始化。

(26)C) 【解析】指针变量 P 存储类型为指针类型,A)选项为字符串类型,B)选项为字符类型,D)选项语法

错误。故选 C)。

(27)A) 【解析】程序开始,  $*p = a$ , 表示把数组  $a$  的第一个元素的地址赋给指针变量  $p$ , 对  $p$  进行初始化。For 循环中  $p++$  表示每执行一次循环, 地址加 1, 即  $p$  指向数组  $a$  的下一个元素, 并通过  $a[i] = *p$  把对应的数组元素赋给  $a[i]$ 。这样第一次执行循环, 得到的是  $a[0] = 10$ ; 第二次得到的是  $a[1] = 20$ , 第三次得到的是  $a[2] = 30$ , 第四次得到的是  $a[3] = 40$ 。

(28)B) 【解析】 $x[N][N]$  的值赋值为  $x[0][0] = 1, x[0][1] = 2, x[0][2] = 3; x[1][0] = 4, x[1][1] = 5, x[1][2] = 6; x[2][0] = 7, x[2][1] = 8, x[2][2] = 9$ 。调用  $\text{fun}()$  过程,  $i$  取值为 0 时,  $j$  取值范围为 1, 2 所以  $b[i] = a[i][j] = 3$ 。

(29)C) 【解析】 $\text{strcpy}(\text{char} * \text{dest}, \text{const char} * \text{src})$  作用是将  $\text{src}$  中的字符串拷贝到以  $\text{dest}$  开始的地址空间。该例中  $\text{strcpy}$  执行后  $a$  的值变为 “a2\0”;  $\text{strcat}(\text{char} * \text{dest}, \text{char} * \text{src})$  作用是把  $\text{src}$  所指字符串添加到  $\text{dest}$  结尾处 (覆盖  $\text{dest}$  结尾处的 '\0') 并添加 '\0',  $\text{strcat}$  执行后  $a$  的值为 “a2yz\0”。

(30)A) 【解析】B) 选项不能将字符串赋值给字符变量; C) 选项中  $\text{name}$  为地址, 不能为其赋值为字符串类型值; D) 选项中 “=” 左面的数组长度小于右面数组的长度。

(31)C) 【解析】 $*s$  为  $\text{char}$  类型指针,  $s[k+1]$  每次执行会后移一位, 第一次执行,  $k++$  后  $k=1$ , 则  $a=0+2=2$ ; 第二次执行  $k=2$ , 进入 if 分支  $a=2+1+1=4$ ; 第三次执行  $k=3, a=4+3=7$ ; 第四次执行  $k=4$ , 进入 if 分支,  $a=7+5=12$ ;

(32)A) 【解析】初始化  $a[0][0] = 'o'; a[1][0] = 't'$ ; 当循环  $i=1, j=3$  时  $a[1][0] = 't', a[3][0] = 'f', a[i][0] > a[j][0]$ , 执行  $a[i][0] = a[j][0]$ , 交换后  $a[1] = \text{“fwo”}$ 。

(33)B) 【解析】此题主要考查函数的调用。 $a=1, b=2$  是赋初值, 程序首先调用  $\text{fun1}$ , 并将实参  $a=5, b=6$  传入到  $\text{fun1}$  中, 此时输出  $a=5, b=6$ 。然后调用  $\text{fun2}$ , 此时  $a=3, b=4$ , 然后调用  $\text{printf}$  打印输出。

(34)A) 【解析】此题主要考查的是静态变量的使用。主函数首先调用  $\text{func}(3), \text{num} = \text{num} + n = 1 + 3 = 4$ , 输出  $\text{num} = 4$ ; 然后调用  $\text{func}(4)$ , 由于  $\text{num}$  是静态变量, 它保存的是上一次的值 4, 所以,  $\text{num} = \text{num} + n = 4 + 4 = 8$ , 输出  $\text{num} = 8$ 。

(35)C) 【解析】由于 C 语言中函数传值方式为实参向形参的单向传值, 而形参的变化并不影响实参, 因此  $\text{fun}$  中不管如何改变  $s$  的值,  $\text{main}$  中的  $q$  始终指向变量  $a$ , 因此  $q$  的值不变。

(36)D) 【解析】此题主要考查的是结构体的使用。程序首先定义并初始化含有 2 个 STU 结构体成员的数组  $c[2]$ , 然后调用子函数  $f, c[1] = b = \{ \text{“Zhao”, 'm', 85, 90} \}$ , 此时  $c[2] = \{ \{ \text{‘Qian’, ‘f’, 95, 92} \}, \{ \text{‘Zhao’, ‘m’, 85, 90} \} \}$ , 最后使用  $\text{printf}$  分别打印输出  $c[2]$  两个结构体的各成员变量。

(37)B) 【解析】 $\text{typedef}$  说明的新类型名习惯上使用大写字母, 但不是必须。所以选项 B) 的说法错误。

(38)A) 【解析】函数的返回值类型可以是简单类型, 也可以是结构体类型, 还可以是指向结构体变量的指针。所以选项 A) 的说法是错误的。

(39)B) 【解析】此题考查的是 C 语言中的关系运算符、逻辑运算符等的使用。求表达式  $(b < 2) / (3 || b)$  的值, 先求被除数  $b < 2$  的值为  $2 * 22 = 8$ , 再求除数  $3 || b$  的值为 1, 因为  $||$  为或关系, 只有两边的值有一个是非 0 值, 那么  $3 || b$  的值为 1。所以表达式  $(b < 2) / (3 || b)$  的值为  $8 / 1 = 8$ 。

(40)D) 【解析】此题考查的是文件操作函数的使用。程序首先将数组  $a[6]$  依次输出到文件  $d2.dat$  中, 接着使用  $\text{rewind}$  函数将文件内部的位置指针指向文件的开头处, 然后将数组  $a[6]$  的元素倒过来写入文件  $d2.dat$  中, 所以最后输出的为 6, 5, 4, 3, 2, 1。

## 二、填空题

(1) 6 或 23

【解析】根据二叉树的性质 3: 对任何一棵二叉树, 度为 0 的结点 (即叶子结点) 总是比度为 2 的结点多一个。因此由度为 2 的结点数可以得出度为 0 的结点数  $= 23 + 1 = 24$ , 又因为此二叉树共有 47 个结点, 所以此二叉树中只有度为 0 的结点和度为 2 的结点。

如果此二叉树是完全二叉树, 则由二叉树性质 4: 具有  $n$  个结点的完全二叉树的深度至少为  $\lceil \log_2 n \rceil + 1$ , 得出此二叉树的深度为 6; 如果此二叉树不是完全二叉树, 而是一个特殊的二叉树 (左单支树或右单支树), 则此二叉树的深度为 23; 故答案为 6 或 23。

(2) 20

【解析】往栈中插入一个元素称为入栈, 入栈操作是首先将栈顶指针加一 (即 Top 加 1), 然后将新元素插入到

栈顶指针指向的位置。从栈中删除一个元素(即删除栈顶元素)称为退栈,退栈操作是首先将栈顶元素(栈顶指针指向的元素)赋给一个指定的变量,然后将栈顶指针减1(即 Top 减1)。栈顶指针 Top 动态反映了栈中元素的变化情况。根据此题中  $\text{top} = 20$ ,可知此栈中元素个数为 20。

(3)逻辑独立性

**【解析】**数据的独立性是指用户的应用程序与数据库中数据是相互独立的,即当数据的物理结构和逻辑结构发生变化时,不影响应用程序对数据的使用。数据的独立性一般分为两种:一种是物理独立性,另一种是逻辑独立性。

物理独立性是指数据的物理结构(包括存储结构、存取方式等)的改变,如存储设备的更换、物理存储的更换、存取方式改变等都不影响数据库的逻辑结构,从而不致引起应用程序的改变。

逻辑独立性是指数据的总体逻辑结构改变时,如修改数据模式、改变数据间的联系等,不需要修改相应的应用程序。所以此处填逻辑独立性。

(4)选择

**【解析】**专门的关系运算包括选择、投影、连接、除等。所以此处填选择。

(5)系统软件

**【解析】**软件根据应用目标的不同,是多种多样的。软件按功能可以分为:应用软件、系统软件、支撑软件(或工具软件)。应用软件是为了解决特定领域的应用而开发的软件。例如,事务处理软件、工程与科学计算软件、实时处理软件、嵌入式软件、人工智能软件等应用性质不同的软件。系统软件是计算机管理自身资源,提高计算机使用效率并为计算机用户提供各种服务的软件。如操作系统、编译程序、汇编程序、网络软件、数据库管理系统等。支撑软件是介于系统软件和应用软件之间,协助用户开发软件的工具性软件,包括辅助、支持开发和维护应用软件的软件,也包括辅助管理人员控制开发进程和项目管理的工具软件,如需求分析工具软件、编码工具软件、测试工具软件、维护工具软件,计划进度管理工具软件,进程控制工具软件等。所以此空应填系统软件。

(6) $a > b$

**【解析】**( ) 的优先级比 ! 高,因此先运算( ) 内的为  $a < = b$ ,然后取非,即为  $a > b$ 。

(7)%lf %f

**【解析】**% 为格式控制起始符,f 标识为单双精度用来输出实数(包括单、双精度),以小数形式输出,不指定宽度,整数部分全部输出并输出 6 位小数。l 标识与 d(整型)在一起指 long 型,与 f(实型)在一起指 double 型。

(8)2 2 4

**【解析】**当输入的字符不是! 时,switch(ch)开始运行,当输入'1'和'3'时,n1 两次实现自增,最后为 2;当输入'2'和'4'时,n2 两次实现自增,最后为 2;当输入'0'、'5'、'6'、'7'时,n3 四次实现自增,最后为 4,故最后输出为 2 2 4。

(9)16

**【解析】**此题使用 for 循环来累加求和,循环体总共执行 4 次,第一次  $\text{sum} = 1$ ,第二次  $\text{sum} = \text{sum} + 3 = 1 + 3 = 4$ ,第三次  $\text{sum} = \text{sum} + 5 = 4 + 5 = 9$ ,第四次  $\text{sum} = \text{sum} + 7 = 9 + 7 = 16$ 。

(10)4321

**【解析】**此题循环 4 次,每次都输出 d 的值。第一次  $d = 1234 \% 10 = 4$ ,  $n = 1234 / 10 = 123$ 。第二次  $d = 123 \% 10 = 3$ ,  $n = 123 / 10 = 12$ 。第三次  $d = 12 \% 10 = 2$ ,  $n = 12 / 10 = 1$ 。第四次  $d = 1 \% 10 = 1$ ,  $n = 1 / 10 = 0$ ,此时不满足循环条件,程序结束。所以,最后的输出结果为 4321。

(11)7

**【解析】**程序调用 st()方法时,\*p=3,p 为 i 的内存地址,进入 st()方法,\*c=7,c 为 k 的内存地址,\*a 为参数接收 p 的指针。故 \*a=3,\*c=7,3>7 返回 false,故直接返回 c 指针赋值给 r,所以 \*r=7。

(12)22

**【解析】**此题考查宏定义的使用。 $x = 2 * (N + M(2)) = 2 * (N + (N + 1) * 2) = 2 * (3 + 4 * 2) = 22$ 。

(13)2

**【解析】**char str[] = "0" 表示的是只有一个 0 元素的字符串,其后还有一个空操作符'\0'隐含着,在内存中各占 1 字节,所以字符串 str 在内存中占用 2 字节。

(14)6

**【解析】**执行 fun(3)后,进入递归算法,n=3,执行 fun(3-1);n=2,执行 fun(2-1);n=1,执行 fun(1-1)返回

1, 返回 fun(2-1)\*n 过程, 值为 2。返回 fun(3-1)\*n 过程, 值为 6。

(15)p -> next

【解析】此题考查链表的使用。While 循环的作用是依次将链表的每个结点的 data 成员打印输出, p 是指向结点地址的指针变量, 这里 p = p -> next 表示输出完一个结点的 data 成员后, 下移到下一个结点的地址。

## 上机真题答案解析

### 一、程序填空题

【参考答案】

(1)x      (2)p      (3)s

【考点分析】

本题考查: 链表的基本操作。了解链表的基本思想和相关算法, 理解有关链表插入及删除时指针移动的先后顺序问题, 注意指针的保存和归位。

【解题思路】

填空 1: 将形参 x 赋值给结点的数据域。

填空 2 和填空 3: 将新的结点和原有链表中的结点进行比较。

【真题链接】

考生可根据本书第 15、28、50 套题进行训练。

### 二、程序改错题

【参考答案】

(1) long k = 1;

(2) num /= 10;

【考点分析】

本题考查: 数据类型, 保存乘积的变量初始化, 除法运算符。

【解题思路】

(1) k 用来存放各位数字的积, 初始值应为 1。

(2) 此处是一个符号错误, 除号用“/”来表示。

【解题宝典】

明确一下 C 语言中的算术运算符:

加法运算符“+”: 为双目运算符, 即应有两个量参与加法运算。

减法运算符“-”: 为双目运算符。但“-”也可作为负值运算符, 此时为单目运算, 例如 -x, -5 等具有左结合性。

乘法运算符“\*”: 为双目运算符, 具有左结合性。

除法运算符“/”: 为双目运算符, 具有左结合性。参与运算量均为整型时, 结果也为整型, 舍去小数。如果运算量中有一个是实型, 则结果为双精度实型。

求余运算符(模运算符)“%”: 为双目运算符, 具有左结合性。要求参与运算的量均为整型。求余运算的结果等于两数相除后的余数。

### 三、程序设计题

【参考答案】

```
float fun (float * a, int n)
{
 float av=0.0;
 int i;
 for(i=0;i<n;i++)/* 求分数的总和*/
 av=av+a[i];
 return (av/n);/* 返回平均值*/
}
```



```
}
```

### 【考点分析】

本题考查:如何通过指针来实现计算平均分。

### 【解题思路】

本题较简单,只需用一个循环语句就可完成数组元素的求和,再将和除以课程数即可。需要注意的是本题对指针的操作,当指针变量指向一个数组时,用该指针变量引用数组元素,引用方式与数组的引用方式相同。例如,本题中 a 指向了 score,所以通过 a 引用 score 中的元素时可以用下标法,也可以用指针运算符,  $a[i]$  和  $*(a+i)$  具有相同的作用。下标运算实际上是从当前地址开始往后取出地址中的第几个元素,当前地址下标为 0。例如,若有 `int cc[10], *p=cc+5;`,即 p 指向了 cc 的第 5 个元素,则 `p[0]` 的作用与 `cc[5]` 相同;`p[3]` 的作用是取出从当前地址(即 p 所指地址)开始往后的第 3 个元素,它与 `cc[8]` 相同;`p[-2]` 的作用是取出从当前地址开始往前的第 2 个元素,它与 `cc[3]` 相同,但不提倡使用“负”的下标。

### 【解题宝典】

计算平均分方法。

掌握以下语句。

```
for (i=0; i<n; i++) /* 求分数的总和 */
 av=av+a[i];
return (av/n); /* 返回平均值 */
```

### 【真题链接】

考生可根据本书第 6、7、27、49、75、96 套填空题进行训练。

# 2012 年 3 月全国计算机等级考试二级 C 语言

## 笔试真题

(考试时间 90 分钟,满分 100 分)

一、选择题((1)~(10)、(21)~(40)每题 2 分,(11)~(20)每题 1 分,共 70 分)

下列各题 A)、B)、C)、D)四个选项中,只有一个选项是正确的。请将正确选项填涂在答题卡相应位置上,答在试卷上不得分。

(1)下列叙述中正确的是( )。

- A)循环队列是队列的一种链式存储结构
- B)循环队列是一种逻辑结构
- C)循环队列是非线性结构
- D)循环队列是队列的一种顺序存储结构

(2)下列叙述中正确的是( )。

- A)栈是一种先进先出的线性表
- B)队列是一种后进先出的线性表
- C)栈与队列都是非线性结构
- D)以上三种说法都不对

(3)一棵二叉树共有 25 个结点,其中 5 个是叶子结点,则度为 1 的结点数为( )。

- A)4
- B)10
- C)6
- D)16

(4)在下列模式中,能够给出数据库物理存储结构与物理存取方法的是( )。

- A)内模式
- B)外模式
- C)概念模式
- D)逻辑模式

(5)在满足实体完整性约束的条件下( )。

- A)一个关系中必须有多个候选关键字
- B)一个关系中只能有一个候选关键字
- C)一个关系中应该有一个或多个候选关键字
- D)一个关系中可以有多个候选关键字

(6)有三个关系 R、S 和 T 如下:

| R |   |   |
|---|---|---|
| A | B | C |
| a | 1 | 2 |
| b | 2 | 1 |
| c | 3 | 1 |

| S |   |   |
|---|---|---|
| A | B | C |
| a | 1 | 2 |
| d | 2 | 1 |

| T |   |   |
|---|---|---|
| A | B | C |
| b | 2 | 1 |
| c | 3 | 1 |

则由关系 R 和 S 得到关系 T 的操作是( )。

- A)差
- B)自然连接
- C)交
- D)并

(7)软件生命周期中的活动不包括( )。

- A)需求分析
- B)市场调研
- C)软件测试
- D)软件维护

(8)下面不属于需求分析阶段任务的是( )。

- A)确定软件系统的功能需求
- B)制定软件集成测试计划
- C)确定软件系统的性能需求
- D)需求规格说明书评审

(9)在黑盒测试方法中,设计测试用例的主要根据是( )。

- A)程序内部逻辑
- B)程序流程图

C) 程序数据结构

D) 程序外部功能

(10) 在软件设计中不使用的工具是( )。

A) 数据流图 (DFD 图)

B) PAD 图

C) 系统结构图

D) 程序流程图

(11) 针对简单程序设计, 以下叙述的实施步骤顺序正确的是( )。

A) 确定算法和数据结构、编码、调试、整理文档

B) 编码、确定算法和数据结构、调试、整理文档

C) 整理文档、确定算法和数据结构、编码、调试

D) 确定算法和数据结构、调试、编码、整理文档

(12) 关于 C 语言中数的表示, 以下叙述中正确的是( )。

A) 只有整型数在允许范围内能精确无误的表示, 实型数会有误差

B) 只要在允许范围内整型数和实型数都能精确的表示

C) 只有实型数在允许范围内能精确无误的表示, 整型数会有误差

D) 只有用八进制表示的数才不会有误差

(13) 以下关于算法的叙述中错误的是( )。

A) 算法可以用伪代码、流程图等多种形式来描述

B) 一个正确的算法必须有输入

C) 一个正确的算法必须有输出

D) 用流程图描述的算法可以用任何一种计算机高级语言编写成程序代码

(14) 以下叙述中错误的是( )。

A) 一个 C 程序中 can 以包含多个不同名的函数

B) 一个 C 程序只能有一个主函数

C) C 程序在书写时, 有严格的缩进要求, 否则不能编译通过

D) C 程序的主函数必须用 main 作为函数名

(15) 设有以下语句:

```
char ch1, ch2; scanf("%c%c", &ch1, &ch2);
```

若要为变量 ch1 和 ch2 分别输入字符 A 和 B, 正确的输入形式应该是( )。

A) A 和 B 之间用逗号间隔

B) A 和 B 之间不能有任何间隔符

C) A 和 B 之间可以用回车间隔

D) A 和 B 之间用空格间隔

(16) 以下选项中非法的字符常量是( )。

A) '\101'

B) '\65'

C) '\xff'

D) '\019'

(17) 有以下程序:

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

```
main()
```

```
{ int a=0, b=0, c=0;
```

```
c = (a - = a - 5); (a = b, b + = 4);
```

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

```
}
```

程序运行后的输出结果是( )。

A) 0,4,5

B) 4,4,5

C) 4,4,4

D) 0,0,0

(18) 设变量均已正确定义并赋值, 以下与其他三组输出结果不同的一组语句是( )。

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

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

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

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

(19)以下选项中,能表示逻辑值“假”的是( )。

- A)1                      B)0.000001                      C)0                      D)100.0

(20)有以下程序:

```
#include <stdio.h>
main()
{
 int a;
 scanf("%d",&a);
 if(a++ < 9)printf("%d\n",a);
 else printf("%d\n",a--);
}
```

程序运行时从键盘输入9<回车>,则输出结果是( )。

- A)10                      B)11                      C)9                      D)8

(21)有以下程序:

```
#include <stdio.h>
main()
{
 int s=0,n;
 for(n=0;n<3;n++)
 {
 switch(s)
 {
 case 0:
 case 1: s+=1;
 case 2: s+=2; break;
 case 3: s+=3;
 default: s+=4;
 }
 printf("%d,",s);
 }
}
```

程序运行后的输出结果是( )。

- A)1,2,4,                      B)1,3,6,                      C)3,10,14,                      D)3,6,10,

(22)若k是int类型变量,且有以下for语句:

```
for(k=-1;k<0;k++)printf("****\n");
```

下面关于语句执行情况的叙述中正确的是( )。

- A)循环体执行一次    B)循环体执行两次    C)循环体一次也不执行    D)构成无限循环

(23)有以下程序:

```
#include <stdio.h>
main()
{
 char a,b,c;
 b='1'; c='A';
 for(a=0;a<6;a++)
 {
 if(a%2) putchar(b+a);
 else putchar(c+a);
 }
}
```

程序运行后的输出结果是( )。

A) 1B3D5F

B) ABCDEF

C) A2C4E6

D) 123456

(24) 设有如下定义语句:

```
int m[] = {2,4,6,8,10}, *k = m;
```

以下选项中,表达式的值为6的是( )。

A) \*(k+2)

B) k+2

C) \*k+2

D) \*k+=2

(25) fun 函数的功能是:通过键盘输入给 x 所指的整型数组所有元素赋值。在下划线处应填写的是( )。

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

```
#define N 5
```

```
void fun(int x[N])
```

```
{ int m;
```

```
for(m = N - 1; m >= 0; m--) scanf("%d", _____);
```

```
}
```

A) &x[++m]

B) &x[m+1]

C) x+(m++)

D) x+m

(26) 若有函数:

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

```
{ }
```

以下叙述中正确的是( )。

A) 调用 fun 函数时只有数组执行按值传送,其他实参和形参之间执行按地址传送

B) 形参 a 和 n 都是指针变量

C) 形参 a 是一个数组名, n 是指针变量

D) 调用 fun 函数时将把 double 型实参数组元素一一对应地传送给形参 a 数组

(27) 有以下程序:

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

```
main()
```

```
{ int a,b,k,m, *p1, *p2;
```

```
k = 1, m = 8;
```

```
p1 = &k, p2 = &m;
```

```
a = / *p1 - m; b = *p1 + *p2 + 6;
```

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

```
}
```

编译时编译器提示错误信息,你认为出错的语句是( )。

A) a = / \*p1 - m;

B) b = \*p1 + \*p2 + 6;

C) k = 1, m = 8;

D) p1 = &k, p2 = &m;

(28) 以下选项中有语法错误的是( )。

A) char \*str[ ] = { "guest" };

B) char str[ ][10] = { "guest" };

C) char \*str[3]; str[1] = "guest";

D) char str[3][10]; str[1] = "guest";

(29) avg 函数的功能是求整型数组中的前若干个元素的平均值,设数组元素个数最多不超过 10,则下列函数说明语句错误的是( )。

A) int avg(int \*a, int n);

B) int avg(int a[10], int n);

C) Int avg(int a, int n);

D) int avg(int a[ ], int n);

(30) 有以下程序:

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

```
#include <string.h>
```

```
main()
{ printf("% d\n",strlen(" ATS\n012\l")); }
```

程序运行后的输出结果是( )。

- A)3                      B)8                      C)4                      D)9

(31)有以下程序:

```
#include <stdio.h>
main()
{ char a[20],b[20],c[20];
 scanf("% s% s", a,b);
 gets(c);
 prind("% s% s% s\n",a,b,c);
}
```

程序运行时从第一列开始输入:

This is a cat! <回车>

则输出结果是( )。

- A)Thisisacat!              B)Thisis a              C)Thisis a cat!              D)Thisisa cat!

(32)有以下程序:

```
#include <stdio.h>
void fun(char c)
{ if(c > 'x') fun(c - 1);
 printf("% c", c);
}
main()
{ fun('z'); }
```

程序运行后的输出结果是( )。

- A)xyz                      B)wxyz                      C)zyxw                      D)zyx

(33)有以下程序:

```
#include <stdio.h>
void func(int n)
{ int i;
 for(i = 0; i < n; i++) printf(" * ");
 printf(" # ");
}
main()
{ func(3); printf(" ????"); func(4); printf(" \n"); }
```

程序运行后的输出结果是( )。

- A) \*\*\*\*\*#                      B) \*\*\*\*\*#  
C) \*\*\*# ???\*                      D) \*\*\*\*\*#

(34)有以下程序:

```
#include <stdio.h>
void fun(int *s)
{ static int j = 0;
 do { s[j] = s[j] + s[j + 1]; } while(++j < 2);
```

```

}
main()
{
 int k,a[10] = {1,2,3,4,5};
 for(k = 1;k < 3;k++) fun(a);
 for(k = 0;k < 5;k++) printf("%d",a[k]);
 printf("\n");
}

```

程序运行后的输出结果是( )。

- A)12345                      B)23445                      C)34756                      D)35745

(35)有以下程序:

```

#include <stdio. h >
#define S(x) (x)*x*2
main()
{
 int k=5, j=2;
 printf("%d,",S(k+j)); printf("%d\n",S((k-j)));
}

```

程序运行后的输出结果是( )。

- A)98,18                      B)39,11                      C)39,18                      D)98,11

(36)有以下程序:

```

#include <stdio. h >
void exch(int t[])
{
 t[0] = t[5]; }
main()
{
 int x[10] = {1,2,3,4,5,6,7,8,9,10},i=0;
 while(i<=4){ exch(&x[i]); i++; }
 for(i=0;i<5;i++) printf("%d",x[i]);
 printf("\n");
}

```

程序运行后输出的结果是( )。

- A)2 4 6 8 10                      B)1 3 5 7 9                      C)1 2 3 4 5                      D)6 7 8 9 10

(37)设有以下程序段:

```

struct MP3
{
 char name[20];
 char color;
 float price;
} std, *ptr;
ptr = &std;

```

若要引用结构体变量 std 中的 color 成员,写法错误的是( )。

- A)std. color                      B)ptr-> color                      C)std-> color                      D)( \*ptr). color

(38)有以下程序:

```

#include <stdio. h >
struct stu
{
 int num; char name[10]; int age;}

```

```

void fun(struct stu *p)
{ printf(" %s\n", p->name); }

main()
{ structstu x[3] = { {01 <, "Zhang", 20}, {02, "Wang", 19}, {03, "Zhao", 18} };
 fun(x + 2);
}

```

程序运行后的输出结果是( )。

- A) Zhang                      B) Zhao                      C) Wang                      D) 19

(39)有以下程序：

```

#include <stdio.h>

main()
{ int a = 12, c;
 c = (a << 2) << 1;
 printf(" %d\n", c);
}

```

程序运行后的输出结果是( )。

- A) 3                      B) 50                      C) 2                      D) 96

(40)以下函数不能用于向文件中写入数据的是( )。

- A) fteell                      B) fwrite                      C) fputc                      D) fprintf

## 二、填空题(每空 2 分,共 30 分)

请将每空的正确答案写在答题卡【1】~【15】序号的横线上,答在试卷上不得分。

- (1)在长度为 n 的顺序存储的线性表中删除一个元素,最坏情况下需要移动表中的元素个数为 【1】。
- (2)设循环队列的存储空间为 Q(1:30),初始状态为 front = rear = 30。现经过一系列入队与退队运算后,front = 16, rear = 15,则循环队列中有 【2】 个元素。
- (3)数据库管理系统提供的数据库语言中,负责数据的增、删、改和查询的是 【3】。
- (4)在将 E-R 图转换到关系模式时,实体和联系都可以表示成 【4】。
- (5)常见的软件工程方法有结构化方法和面向对象方法,类、继承以及多态性等概念属于 【5】。
- (6)设变量 a 和 b 已定义为 int 类型,若要通过 scanf("a = %d, b = %d", &a, &b); 语句分别给 a 和 b 输入 1 和 2,则正确的数据输入内容是 【6】。
- (7)以下程序的输出结果是 【7】。

```

#include <stdio.h>

main()
{ int a = 37;
 a + = a% = 9; printf(" %d\n", a);
}

```

- (8)设 a、b、c 都是整型变量,如果 a 的值为 1, b 的值为 2,则执行 c = a++||b++; 语句后,变量 b 的值是 【8】。
- (9)有以下程序段：

```

s = 1.0;
for(k = 1; k <= n; k++) s = s + 1.0 / (k * (k + 1));
printf(" %f\n", s);

```

请填写,使以下程序段的功能与上面的程序段完全相同。

```

s = 1.0; k = 1;
while(【9】)

```



```
 { s = s + 1.0 \ (k * (k + 1)); k = k + 1; }
```

```
printf("%f\n", s);
```

(10) 以下程序的输出结果是 【10】。

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

```
main()
```

```
{ char a, b;
```

```
for(a = 0; a < 20; a += 7) { b = a % 10; putchar(b + '0'); }
```

```
}
```

(11) 以下程序的输出结果是 【11】。

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

```
main()
```

```
{ char *ch[4] = { "red", "green", "blue" };
```

```
int i = 0;
```

```
while(ch[i])
```

```
{ putchar(ch[i][0]); i++; }
```

```
}
```

(12) 有以下程序：

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

```
main()
```

```
{ int arr[] = { 1, 3, 5, 7, 2, 4, 6, 8 }, i, start;
```

```
scanf("%d", &start);
```

```
for(i = 0; i < 3; i++)
```

```
printf("%d", arr[(start + i) % 8]);
```

```
}
```

若在程序运行时输入整数 10 <回车>，则输出结果为 【12】。

(13) 以下程序的功能是输出 a 数组中所有字符串，请填空。

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

```
main()
```

```
{ char *a[] = { "ABC", "DEFGH", "IJ", "KLMNOP" };
```

```
int i = 0;
```

```
for(; i < 4; i++) printf("%s\n", 【13】); }
```

(14) 以下程序的输出结果是 【14】。

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

```
#include <stdlib.h>
```

```
#include <string.h>
```

```
main()
```

```
{ char *p, *q, *r;
```

```
p = q = r = (char *) malloc(sizeof(char) * 20);
```

```
strcpy(p, "attaboy, welcome!");
```

```
printf("%c%c%c\n", p[11], q[3], r[4]);
```

```
free(p);
```

```
}
```

(15) 设文件 test.txt 中原已写入字符串 Begin，执行以下程序后，文件中的内容为 【15】。

```
#include <stdio.h>

main()
{
 FILE *fp;

 fp = fopen("test.txt", "w+");
 fputs("test", fp);
 fclose(fp);
}
```

# 全国计算机等级考试二级 C 语言

## 上机真题

(考试时间 90 分钟,满分 100 分)

### 一、程序填空题

下列给定程序中,函数 fun 的功能是:将形参 a 所指数组中的前半部分元素中的值与后半部分元素中的值对换。形参 n 中存放数组中数据的个数,若 n 为奇数,则中间的元素不动。

例如,若 a 所指数组中的数据为:1,2,3,4,5,6,7,8,9,则调换后为:6,7,8,9,5,1,2,3,4。

请在下划线处填入正确的内容并将下划线删除,使程序得出正确的结果。

注意:部分源程序给出如下。

不得增行或删行,也不得更改程序的结构!

试题程序:

```
#include <stdio.h>

#define N 9

void fun(int a[], int n)
{
 int i, t, p;
 p = (n%2==0)? n/2:n/2+【1】;
 for(i=0;i<n/2;i++)
 {
 t=a[i];
 a[i]=a[p+【2】];
 【3】=t;
 }
}

main()
{
 int b[N]={1, 2, 3, 4, 5, 6, 7, 8, 9},i;
 printf("\nThe original data:\n");
 for (i=0;i<N;i++)
 printf("%4d",b[i]);
 printf("\n");
}
```