

机密★启用前

# 2008 年 9 月全国计算机等级考试二级笔试试卷

## C 语言程序设计

# 24

### 注意事项

- 一、考生应严格遵守考场规则，得到监考人员指令后方可作答。
- 二、考生拿到试卷后应首先将自己的姓名、准考证号等内容涂写在答题卡的相应位置上。
- 三、选择题答案必须用铅笔填涂在答题卡的相应位置上，填空题的答案必须用蓝、黑色钢笔或圆珠笔写在答题卡的相应位置上，答案写在试卷上无效。
- 四、注意字迹清楚，保持卷面整洁。
- 五、考试结束将试卷和答题卡放在桌上，不得带走。待监考人员收毕清点后，方可离场。

\*\*\* 版权所有，任何单位或个人不得保留、复制和出版，违者必究 \*\*\*

教育部考试中心

二 00 八年七月制

# 2008 年 9 月全国计算机等级考试笔试试卷

## 二级公共基础知识和 C 语言程序设计

(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, 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) 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
- (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) 笛卡尔积      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 语句必须在一行内写完
- (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 语言表达式是 ( )。
- 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) 若有表达式(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 (o 代表一个空格符) ( )。
- A) o o o 1,2,3<回车>                      B) 1 o 2 o 3<回车>
- C) 1,o o o 2,o o o 3<回车>                      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\n", 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=50 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前
- 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");
  }
}
```

程序的运行结果是 ( )。

- |          |          |
|----------|----------|
| A) 2 3 4 | B) 4 3 2 |
| 3 4 5    | 5 4 3    |
| C) 2 3   | D) 4 5   |
| 3 4      | 3 4      |
| 4 5      | 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++);
}
```

程序的运行结果是 ( )。

- |        |       |      |      |
|--------|-------|------|------|
| A) 331 | B) 41 | 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("i=%d", i); break;}
    i++;
  } while(i!=0);
  printf("\n");
}
```

程序的运行结果是 ( )。

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));}
程序的运行结果是 ( )。
```

A) 1

B) 2

C) 3

D) 4

(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));
}
程序的运行结果是 ( )。
```

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");
}
程序的运行结果是 ( )。
```

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);
}

```

程序的运行结果是 ( )。

- 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}, i;
  fun(k,5);
  for(i=2; i<8; i++) printf("%d", k[i]);
  printf("\n");
}

```

程序的运行结果是 ( )。

- A) 345678                      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");
}

```

程序的运行结果是 ( )。

- A) 1,2,3,4,                      B) 1,0,7,0,                      C) 1,4,5,9,                      D) 3,4,8,10,

(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));
}

```

程序的运行结果是 ( )。

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

(37) 有以下程序

```

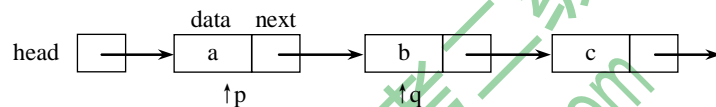
#include <stdio.h>
main()
{
    struct STU { char name[9]; char sex; double score[2]; };
    struct STU a={"Zhao",'m',85.0,90.0}, b={"Qian",'f',95.0,92.0};
    b=a;
    printf("%s,%c,%2.0f,%2.0f\n",b.name,b.sex,b.score[0],b.score[1]);
}

```

程序的运行结果是 ( )。

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

(38) 假定已建立以下链表结构，且指针 p 和 q 已指向如图所示的结点：



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组是 ( )。

- A) (\*p).next=(\*q).next; free(p);              B) p=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);
}

```

程序的运行结果是 ( )。

- 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);
}

```

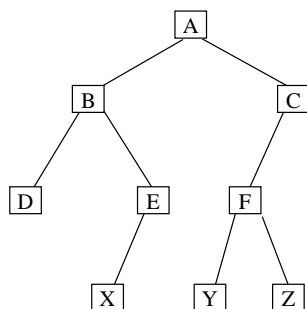
以上程序执行后 abc.dat 文件的内容是 ( )。

- B) China                      B) Chinang                      C) ChinaBeijing                      D) BeijingChina

## 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) 对下列二叉树进行中序遍历的结果 【1】。



- (2) 按照软件测试的一般步骤, 集成测试应在 【2】 测试之后进行。
- (3) 软件工程三要素包括方法、工具和过程, 其中, 【3】 支持软件开发的各个环节的控制和管理。
- (4) 数据库设计包括概念设计、【4】 和物理设计。
- (5) 在二维表中, 元组的 【5】 不能再分成更小的数据项。
- (6) 设变量 a 和 b 已正确定义并赋初值。请写出与  $a=a+b$  等价的赋值表达式 【6】。
- (7) 若整型变量 a 和 b 中的值分别为 7 和 9, 要求按以下格式输出 a 和 b 的值:

a=7

b=9

请完成输出语句: `printf(" 【7】 ",a,b);`。

- (8) 以下程序的输出结果是 【8】。

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

- (9) 以下程序的输出结果是 【9】。

```
#include <stdio.h>
main()
{   int  j, a[]={1,3,5,7,9,11,13,15},*p=a+5;
    for(j=3; j; j--)
    {   switch(j)
        {   case 1:
            case 2: printf("%d",*p++); break;
            case 3: printf("%d",*(--p));
        }
    }
}
```



(10) 以下程序的输出结果是 **【10】**。

```
#include <stdio.h>
#define N 5
int fun(int *s, int a, int n)
{ int j;
  *s=a; j=n;
  while(a!=s[j])j--;
  return j;
}
main()
{ int s[N+1]; int k;
  for(k=1; k<=N; k++) s[k]=k+1;
  printf("%d\n", fun(s, 4, N));
}
```

(11) 以下程序的输出结果是 **【11】**。

```
#include <stdio.h>
int fun(int x)
{ static int t=0;
  return(t +=x);
}
main()
{ int s,i;
  for(i=1; i<=5; i++) s=fun(i);
  printf("%d\n", s);
}
```

(12) 以下程序按下面指定的数据给 x 数组的下三角置数，并按如下形式输出，请填写。

```
4
3 7
2 6 9
1 5 8 10
```

```
#include <stdio.h>
main()
{ int x[4][4], n=0, i, j;
  for(j=0; j<4; j++)
    for(i=3; i>=j; 【12】) { n++; x[i][j]=【13】; }
  for(i=0; i<4; i++)
  { for(j=0; j<=i; j++) printf("%3 d", x[i][j]);
    printf("\n");
  }
}
```

(13) 以下程序的功能是：通过函数 func 输入字符并统计输入字符的个数。输入时用字符 @ 作为输入结束标志。请填写。

```
#include <stdio.h>
long 【14】 ; /* 函数说明语句 */
```

```
main()
{ long n;
  n=func(); printf("n=%ld\n",n);
}
long func()
{ long m;
  for( m=0; getchar()!='@'; 【15】);
  return m;
}
```

考一级就上考一级网  
www.kaoerji.com