

2009 年 3 月全国计算机等级考试二级笔试试卷

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, 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, 请将正确选项涂写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) 栈是“先进先出”的线性表
 - B) 队列是“先进后出”的线性表
 - C) 循环队列是非线性结构
 - D) 有序线性表既可以采用顺序存储结构, 也可以采用链式存储结构
- (2) 支持子程序调用的数据结构是 ()。
 - A) 栈
 - B) 树
 - C) 队列
 - D) 二叉树
- (3) 某二叉树有 5 个度为 2 的结点, 则该二叉树中的叶子结点数是 ()。
 - A) 10
 - B) 8
 - C) 6
 - D) 4
- (4) 下列排序方法中, 最坏情况下比较次数最少的是 ()。
 - A) 冒泡排序
 - B) 简单选择排序
 - C) 直接插入排序
 - D) 堆排序
- (5) 软件按功能可以分为: 应用软件、系统软件和支撑软件(或工具软件)。下面属于应用软件的是 ()。
 - A) 编译程序
 - B) 操作系统
 - C) 教务管理系统
 - D) 汇编程序
- (6) 下面叙述中错误的是 ()。
 - A) 软件测试的目的是发现错误并改正错误
 - B) 对被调试的程序进行“错误定位”是程序调试的必要步骤
 - C) 程序调试通常也称为 Debug
 - D) 软件测试应严格执行测试计划, 排除测试的随意性
- (7) 耦合性和内聚性是对模块独立性度量的两个标准。下列叙述中正确的是 ()。
 - A) 提高耦合性降低内聚性有利于提高模块的独立性
 - B) 降低耦合性提高内聚性有利于提高模块的独立性
 - C) 耦合性是指一个模块内部各个元素间彼此结合的紧密程度
 - D) 内聚性是指模块间互相连接的紧密程度
- (8) 数据库应用系统中的核心问题是 ()。
 - A) 数据库设计
 - B) 数据库系统设计
 - C) 数据库维护
 - D) 数据库管理员培训
- (9) 有两个关系 R, S 如下:

R		
A	B	C
a	3	2
b	0	1
c	2	1

S	
A	B
a	3
b	0
c	2

由关系 R 通过运算得到关系 S, 则所使用的运算为 ()。

- (10) 将 E-R 图转换为关系模式时, 实体和联系都可以表示为 ()。

- A) 属性
C) 关系
B) 键
D) 域

(11) 以下选项中合法的标识符是 ()。

- A) 1_1
C) _11
B) 1—1
D) 1_ _

(12) 若函数中有定义语句: int k; 则 ()。

- A) 系统将自动给 k 赋初值 0
C) 系统将自动给 k 赋初值 -1
B) 这时 k 中的值无定义
D) 这时 k 中无任何值

(13) 以下选项中, 能用作数据常量的是 ()。

- A) 0115
C) 1.5e1.5
B) 0118
D) 115L

(14) 设有定义: int x=2; 以下表达式中, 值不为 6 的是 ()。

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

(15) 程序段: int x=12; double y=3.141593; printf("%d%8.6f",x,y); 的输出结果是 ()。

- A) 123.141593
C) 12, 3.141593
B) 12 3.141593
D) 123.1415930

(16) 若有定义语句: double x, y, *px, *py; 执行了 px=&x; py=&y; 之后, 正确的输入语句是 ()。

- A) scanf("%f%f", x, y);
C) scanf("%lf%le", px, py);
B) scanf("%f%f" &x, &y);
D) scanf("%lf%lf", x, y);

(17) 以下是 if 语句的基本形式:

if(表达式) 语句

其中“表达式”

- A) 必须是逻辑表达式
C) 必须是逻辑表达式或关系表达式
B) 必须是关系表达式
D) 可以是任意合法的表达式

(18) 有以下程序

```
#include <stdio.h>
main()
{ int x;
  scanf ("%d", &x);
  if(x<=3); else
  if (x!=10) printf("%d\n", x);
}
```

程序运行时, 输入的值在哪个范围才会有输出结果 ()。

- A) 不等于 10 的整数
C) 大于 3 或等于 10 的整数
B) 大于 3 且不等 10 的整数
D) 小于 3 的整数

(19) 有以下程序

```
#include <stdio.h>
main()
{ int a=1,b=2,c=3,d=0;
  if(a==1&&b++==2)
  if(b!=2 || c--!=3)
    printf("%d,%d,%d\n",a,b,c);
  else printf("%d,%d,%d\n",a,b,c);
  else printf("%d,%d,%d\n",a,b,c);
}
```

程序运行后的输出结果是 ()。

A) 1,2,3

B) 1,3,2

C) 1,3,3

D) 3,2,1

(20) 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) *

(21) 有以下程序

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

```
main()
```

```
{ char *s={"ABC"};
```

```
    do
```

```
    { printf("%d",*s%10); s++;
```

```
    } while(*s);
```

```
}
```

注意: 字母 A 的 ASCII 码值为 65。程序运行后的输出结果是 ()。

A) 5670

B) 656667

C) 567

D) ABC

(22) 设变量已正确定义, 以下不能统计出一行中输入字符个数 (不包含回车符) 的程序段是 ()。

A) n=0; while((ch=getchar())!='\n')n++;

B) n=0;while(getchar()!='\n')n++;

C) for(n=0; getchar()!='\n'; n++);

D) n=0; for(ch=getchar(); ch!='\n'; n++);

(23) 有以下程序

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

```
main()
```

```
{ int a1,a2; char c1,c2;
```

```
    scanf("%d%c%d%c",&a1,&c1,&a2,&c2);
```

```
    printf("%d,%c,%d,%c",a1,c1,a2,c2);
```

```
}
```

若想通过键盘输入, 使得 a1 的值为 12, a2 的值为 34, c1 的值为字符 a, c2 的值为字符 b, 程序输出结果是: 12,a,34,b 则正确的输入格式是 (以下 □ 代表空格, <CR>代表回车) ()。

A) 12a34b<CR>

B) 12 □ a □ 34 □ b<CR>

C) 12,a,34,b<CR>

D) 12 □ a34 □ b<CR>

(24) 有以下程序

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

```
int f(int x,int y)
```

```
{return ((y-x)*x);}
```

```
main()
```

```
{ int a=3,b=4,c=5,d;
```

```
    d=f(f(a,b),f(a,c));
```

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

```
}
```

程序运行后的输出结果是 ()。

A) 10

B) 9

C) 8

D) 7

(25) 有以下程序

```
#include <stdio.h>

void fun(char *s)
{
    while(*s)
    {
        if(*s%2==0)printf("%c",*s);
        s++;
    }
}

main()
{
    char a[]={ "good" };
    fun(a);printf("\n");
}
```

注意：字母 a 的 ASCII 码值为 97，程序运行后的输出结果是（ ）。

- A) d B) go
C) god D) good

(26) 有以下程序

```
#include <stdio.h>

void fun(int *a,int *b)
{   int *c;
    c=a;a=b;b=c;
}

main()
{   int x=3,y=5,*p=&x,*q=&y;
    fun(p,q);printf("%d,%d",*p,*q);
    fun(&x,&y);printf("%d,%d\n",*p,*q);
}
```

程序运行后的输出结果是 ()。

- A) 3,5,5,3 B) 3,5,3,5
C) 5,3,3,5 D) 5,3,5,3

(27) 有以下程序

```
#include <stdio.h>

void f(int *p,int *q);

main()
{   int m=1,n=2,*r=&m;
    f(r,&n);printf("%d,%d",m,n);
}

void f(int *p,int *q)
{   p=p+1;*q=*q+1; }

程序运行后的输出结果是（
```

程序运行后的输出结果是 ()。

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

(28) 以下函数按每行 8 个输出数组中的数据

```
void fun ( int *w, int n)
{   int i;
    for (i=0; i<n; i++)
    { _____
```

```

    printf ("%d", w[i]);
}
printf ("\n");
}

```

下划线处应填入的语句是 ()。

- A) if(i/8==0)printf("\n");
 B) if(i/8=0)continue;
 C) if(i%8==0)printf("\n");
 D) if(i%8==0)continue;

(29) 若有以下定义

```
int x[10], *pt=x;
```

则对 x 数组元素的正确引用是 ()。

- A) *&x[10]
 B) *(x+3)
 C) *(pt+10)
 D) pt+3

(30) 设有定义: char s[8]; 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';

(31) 有以下程序

```

#include <stdio.h>
main()
{
    char *a[]={ "abcd", "ef", "gh", "ijk" }; int i;
    for(i=0; i<4; i++) printf("%c", *a[i]);
}

```

程序运行后的输出结果是 ()。

- A) aegi
 B) dfhk
 C) abcd
 D) abcdefghijk

(32) 以下选项中正确的语句组是 ()。

- A) char s[]; s="BOOK!";
 B) char *s; s={"BOOK!";
 C) char s[10]; s="BOOK!";
 D) char *s; s="BOOK!";

(33) 有以下程序

```

#include <stdio.h>
int fun(int x,int y)
{
    if(x==y)return(x);
    else return((x+y)/2);
}
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) 3
 B) 6
 C) 8
 D) 12

(34) 设函数中有整型变量 n, 为保证其在未赋初值的情况下初值为 0, 应选择的存储类别是 ()。

- A) auto
 B) register
 C) static
 D) auto 或 register

(35) 有以下程序

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


若文本文件 filea.txt 中原有内容为：hello，则运行以上程序后，文件 filea.txt 中的内容为（ ）。

```
#include <stdio.h>
main()
{   char a=0,ch;
    while((ch=getchar())!='\n')
```



```

    {   if(a%2!=0&&(ch>='a'&&ch<='z')) ch=ch-'a'+'A';
        a++; putchar(ch);
    }
    printf("\n");
}

```

(11) 有以下程序，程序执行后，输出结果是_____。

```

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

```

(12) 请将以下程序中的函数声明语句补充完整。

```

#include <stdio.h>
int _____;
main()
{   int x, y, (*p)();
    scanf("%d%d", &x, &y);
    p=max;
    printf("%d\n", (*p)(x,y));
}
int max(int a, int b)
{ return(a>b?a:b); }

```

(13) 以下程序用来判断指定文件是否能正常打开，请填空。

```

#include <stdio.h>
main()
{   FILE *fp;
    if(((fp=fopen("test.txt", "r"))== _____))
        printf("未能打开文件!\n");
    else
        printf("文件打开成功!\n");
}

```

(14) 下列程序的运行结果为_____。

```

#include <stdio.h>
#include <string.h>
struct A
{   int a; char b[10]; double c;};
void f(struct A *t);
main()
{   struct A a={ 1001,"ZhangDa",1098.0};
    f(&a);printf("%d,%s,%6.1f\n",a.a,a.b,a.c);
}
void f(struct A *t)

```



```
{ strcpy(t->b,"ChangRong"); }
```

- (15) 以下程序把三个 NODETYPE 型的变量链接成一个简单的链表，并在 while 循环中输出链表结点数据域中的数据。请填空。

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

```
struct node
```

```
{ int data; struct node *next;};
```

```
typedef struct node NODETYPE;
```

```
main()
```

```
{ NODETYPE a,b,c *h,*p;
```

```
  a.data=10; b.data=20; c.data=30; h=&a;
```

```
  a.next=&b; b.next=&c; c.next='\0';
```

```
  p=h;
```

```
  while (p) { printf("%d,", p->data); _____; }
```

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

```
}
```

