

机密★启用前

2005 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 五年七月制

2005 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 120 分钟, 满分 100 分)

一、选择题 ((1) - (10) 每小题 2 分, (11) - (50) 每小题 1 分, 共 60 分)

- (1) 下列叙述中正确的是 ()。
- A) 程序设计就是编制程序
 - B) 程序的测试必须由程序员自己去完成
 - C) 程序经调试改错后还应进行再测试
 - D) 程序经调试改错后不必进行再测试
- (2) 下列数据结构中, 能用二分法进行查找的是 ()。
- A) 顺序存储的有序线性表
 - B) 线性链表
 - C) 二叉链表
 - D) 有序线性链表
- (3) 下列关于栈的描述正确的是 ()。
- A) 在栈中只能插入元素而不能删除元素
 - B) 在栈中只能删除元素而不能插入元素
 - C) 栈是特殊的线性表, 只能在一端插入或删除元素
 - D) 栈是特殊的线性表, 只能在一端插入元素, 而在另一端删除元素
- (4) 下列叙述中正确的是 ()。
- A) 一个逻辑数据结构只能有一种存储结构
 - B) 数据的逻辑结构属于线性结构, 存储结构属于非线性结构
 - C) 一个逻辑数据结构可以有多种存储结构, 且各种存储结构不影响数据处理的效率
 - D) 一个逻辑数据结构可以有多种存储结构, 且各种存储结构影响数据处理的效率
- (5) 下列描述中正确的是 ()。
- A) 软件工程只是解决软件项目的管理问题
 - B) 软件工程主要解决软件产品的生产率问题
 - C) 软件工程的主要思想是强调在软件开发过程中需要应用工程化原则
 - D) 软件工程只是解决软件开发中的技术问题
- (6) 在软件设计中, 不属于过程设计工具的是 ()。
- A) PDL (过程设计语言)
 - B) PAD 图
 - C) N-S 图
 - D) DFD 图
- (7) 下列叙述中正确的是 ()。
- A) 软件交付使用后还需要再进行维护
 - B) 软件工具交付使用就不需要再进行维护
 - C) 软件交付使用后其生命周期就结束
 - D) 软件维护是指修复程序中被破坏的指令
- (8) 数据库设计的根本目标是要解决 ()。
- A) 数据共享问题
 - B) 数据安全问题
 - C) 大量数据存储问题
 - D) 简化数据维护
- (9) 设有如下关系表:

R		
A	B	C
1	1	2
2	2	3

S		
A	B	C
3	1	3

T		
A	B	C
1	1	2
2	2	3
3	1	3

则下列操作中正确的是 ()。

- A) $T=R \cap S$ B) $T=R \cup S$ C) $T=R \times S$ D) $T=R/S$

(10) 数据库系统的核心的是 ()。

- A) 数据模型 B) 数据库管理系统
C) 数据库 D) 数据库管理员

(11) 下列叙述中错误的是 ()。

- A) 用户所定义的标识符允许使用关键字
B) 用户所定义的标识符应尽量做到“见名知意”
C) 用户所定义的标识符必须以字母或下划线开头
D) 用户定义的标识符中、大、小写字母代表不同标识

(12) 下列叙述中错误的是 ()。

- A) C 语句必须以分号结束
B) 复合语句在语法上被看作一条语句
C) 空语句出现在任何位置都不会影响程序运行
D) 赋值表达式末尾加分号就构成赋值语句

(13) 下列叙述中正确的是 ()。

- A) 调用 printf()函数时，必须要有输出项
B) 使用 putchar()函数时，必须在之前包含头文件 stdio.h
C) 在 C 语言中，整数可以以二进制、八进制或十六进制的形式输出
D) 调节 getchar()函数读入字符时，可以从键盘上输入字符所对应的 ASCII 码

(14) 下列关于函数的叙述中正确的是 ()。

- A) 每个函数都可以被其他函数调用 (包括 main 函数)
B) 每个函数都可以被单独编译
C) 每个函数都可以单独运行
D) 在一个函数内部可以定义另一个函数

(15) 若有语句: char *line[5]; 下列叙述中正确的是 ()。

- A) 定义 line 是一个数组，每个数组元素是一个基类型为 char 的指针变量
B) 定义 line 是一个指针变量，该变量可以指向一个长度为 5 的字符型数组
C) 定义 line 是一个指针数组，语句中的*号称为间址运算符
D) 定义 line 是一个指向字符型函数的指针

(16) 有下列程序段:

```
typedef struct NODE
{ int num; struct NODE *next;
} OLD;
```

下列叙述中正确的是 ()。

- A) 以上的说明形式非法 B) NODE 是一个结构体类型
C) OLD 是一个结构体类型 D) OLD 是一个结构体变量

(17) 下列叙述中错误的是 ()。

- A) 在 C 语言中，对二进制文件的访问速度比文本文件快

- B) 在 C 语言中, 随机文件以二进制代码形式存储数据
 C) 语句 FILE fp; 定义了一个名为 fp 的文件指针
 D) C 语言中的文本文件以 ASC II 码形式存储数据
- (18) 当把四个表达式用作 if 语句的控制表达式时, 有一个选项与其他三个选项含义不同, 这个选项是 ()。
- A) k%2
 B) k%2==1
 C) (k%2)!=0
 D) !k%2==1
- (19) 下列不能正确计算代数式 $\frac{1}{3}\sin^2\left(\frac{1}{2}\right)$ 值的 C 语言表达式是 ()。
- A) 1/3 *sin(1/2)*sin(1/2)
 B) sin(0.5)*sin(0.5)/3
 C) pow(sin(0.5),2)/3
 D) 1/3.0*pow(sin(1.0/2),2)
- (20) 下列能正确定义且赋初值的语句是 ()。
- A) int n1=n2=10;
 B) char c=32;
 C) float f=f+1.1;
 D) double x=12.3E2.5;
- (21) 下列程序的功能是: 给 r 输入数据后计算半径为 r 的圆面积 s。程序在编译时出错。
- ```
main()
/* Beginning */
{ int r; float s;
 scanf("%d",&r);
 s=* π *r*r;
 printf("s=%f\n",s);
}
```
- 出错的原因是 ( )。
- A) 注释语句书写位置错误  
 B) 存放圆半径的变量 r 不应该定义为整型  
 C) 输出语句中格式描述符非法  
 D) 计算圆面积的赋值语句中使用了非法变量
- (22) 设有定义: int k=1,m=2; float f=7;则下列选项中错误的表达式是 ( )。
- A) k=k>=k  
 B) -k++  
 C) k%int(f)  
 D) k>=f>=m
- (23) 设有定义: int a=2,b=3,c=4;则下列选项中值为 0 的表达式是 ( )。
- A) (! a==1)&&(!b==0)  
 B) (a<b)&&! c || 1  
 C) a && b  
 D) a || (b+b)&&(c-a)
- (24) 有下列程序段:
- ```
int k=0,a=1,b=2,c=3;
k=a<b?b:a; k=k>c?c:k;
```
- 执行该程序段后, k 的值是 ()。
- A) 3
 B) 2
 C) 1
 D) 0
- (25) 设变量 a、b、c、d 和 y 都已正确定义并赋值。若有下列 if 语句
- ```
if(a<b)
 if(c==d) y=0;
 else y=1;
```
- 该语句所表示的含义是 ( )。

$$A) y = \begin{cases} 0 & a < b \text{ 且 } c = d \\ 1 & a \geq b \end{cases}$$

$$B) y = \begin{cases} 0 & a < b \text{ 且 } c = d \\ 1 & a \geq b \text{ 且 } c \neq d \end{cases}$$

$$C) y = \begin{cases} 0 & a < b \text{ 且 } c = d \\ 1 & a < b \text{ 且 } c \neq d \end{cases}$$

$$D) y = \begin{cases} 0 & a < b \text{ 且 } c = d \\ 1 & c \neq d \end{cases}$$

(26) 有下列程序段

```
int n,t=1,s=0;
scanf("%d",&n);
do{ s=s+t;t=t-2; }while(t!=n);
```

为使此程序段不陷入死循环, 从键盘输入的数据应该是 ( )。

- A) 任意正奇数                                      B) 任意负偶数  
C) 任意正偶数                                      D) 任意负奇数

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

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

(28) 设有定义:  $\text{int } n1=0,n2,*p=\&n2,*q=\&n1;$ , 以下赋值语句中与  $n2=n1$ ; 语句等价的是 ( )。

- A)  $*p=*q;$                                       B)  $p=q;$                                       C)  $*p=\&n1;$                                       D)  $p=*q;$

(29) 若有定义:  $\text{int } x=0,*p=\&x;$ , 则语句  $\text{printf}("%d\n",*p);$  的输出结果是 ( )。

- A) 随机值                                      B) 0                                      C) x 的地址                                      D) p 的地址

(30) 设 fun() 函数的定义形式为 ( )。

```
void fun(char ch,float x){...}
```

则下列对函数 fun 的调用语句中, 正确的是

- A) fun("abc",3.0);                                      B) t=fun('D',16.5);  
C) fun('65',2.8);                                      D) fun(32,32);

(31) 有下列程序:

```
main()
{ int a[10]={1,2,3,4,5,6,7,8,9,10},*p=&a[3],*q=p+2;
 printf("%d\n",*p+*q);
}
```

程序运行后的输出结果是 ( )。

- A) 16                                      B) 10                                      C) 8                                      D) 6

(32) 有下列程序:

```
main()
{ char p[]={'a','b','c'},q[]="abc";
 printf("%d%d\n",sizeof(p),sizeof(q));
};
```

程序运行后的输出结果是 ( )。

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

(33) 有下列程序:

```
#define f(x)=(x*x)
```

```

main()
{ int i1,i2;
 i1=f(8)/f(4); i2=f(4+4)/f(2+2);
 printf("%d, %d\n",i1,i2);
}

```

程序运行后的输出结果是 ( )。

- A) 64,28                      B) 4,4                      C) 4,3                      D) 64,64

(34) 有下列程序:

```

main()
{ char a1='M',a2='m';
 printf("%c\n",(a1,a2));
}

```

下列叙述中正确的是 ( )。

- A) 程序输出大写字母 M                      B) 程序输出小写字母 m  
 C) 格式说明符不足, 编译出错                      D) 程序运行时产生出错信息

(35) 有下列程序:

```

#include <stdio.h>
main()
{ char c1='1',c2='2';
 c1=getchar(); c2=getchar(); putchar(c1); putchar(c2);
}

```

当运行时输入:a<CR>后, 下列叙述正确的是 ( )。

- A) 变量 c1 被赋予字符 a,c2 被赋予回车符  
 B) 程序将等待用户输入第 2 个字符  
 C) 变量 c1 被赋予字符 a,c2 中仍是原有字符 2  
 D) 变量 c1 被赋予字符 a, c2 中将无确定值

(36) 有下列程序:

```

main()
{ int k=5,n=0;
 while(k>0)
 { switch(k)
 { default:break;
 case 1: n+=k;
 case 2:
 case 3: n+=k;
 }
 k--;
 }
 printf("%d\n",n);
}

```

程序运行后的输出结果是 ( )。

- A) 0                      B) 4                      C) 6                      D) 7

(37) 有下列程序:

```

main()

```

```

{ int a[]={2,4,6,8,10},y=0,x,*p;
 p=&a[1];
 for(x=1;x<3;x++) y+=p[x];
 printf("%d\n",y);
}

```

程序运行后的输出结果是 ( )。

- A) 10                      B) 11                      C) 14                      D) 15

(38) 有下列程序:

```

void sort(int a[],int n)
{ int i,j,t;
 for(i=0;i<n-1;i++)
 for(j=i+1;j<n;j++)
 if(a[i]<a[j]){ t=a[i]; a[i]=a[j]; a[j]=t;}
}
main()
{ int aa[10]={1,2,3,4,5,6,7,8,9,10},i;
 sort(aa+2,5);
 for(i=0;i<10;i++)printf("%d,",aa[i]);
 printf("\n");
}

```

程序运行后的输出结果是 ( )。

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

(39) 有下列程序:

```

void sum(int a[])
{ a[0]=a[-1]+a[1];
}
main()
{ int a[10]={1,2,3,4,5,6,7,8,9,10};
 sum(&a[2]);
 printf("%d\n",a[2]);
}

```

程序运行后的输出结果是 ( )。

- A) 6                      B) 7                      C) 5                      D) 9

(40) 有下列程序:

```

void swap1 (int c0[],int c1[])
{ int t;
 t=c0[0]; c0[0]=c1[0]; c1[0]=t;
}
void swap2(int *c0,int *c1)
{ int t;
 t=*c0; *c0=*c1; *c1=t;
}
main()

```

```

{ int a[2]={3,5}, b[2]={3,5};
 swap1(a,a+1); swap2(&b[0],&b[1]);
 printf("%d %d %d %d\n",a[0],a[1],b[0],b[1]);
}

```

程序运行后的输出结果是 ( )。

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

(41) 有下列程序:

```

#include<string.h>
main()
{ char p[]={'a', 'b', 'c'},q[10]={ 'a', 'b', 'c'};
 printf("%d%d\n",strlen(p),strlen(q));
}

```

下列叙述中正确的是 ( )。

- A) 在给 p 和 q 数组赋初值时, 系统会自动添加字符串结束符, 故输出的长度都为 3  
 B) 由于 p 数组中没有字符串结束符, 长度不能确定, 但 q 数组中字符串长度为 3  
 C) 由于 q 数组中没有字符串结束符, 长度不能确定, 但 p 数组中字符串长度为 3  
 D) 由于 p 和 q 数组中都没有字符串结束符, 故长度都不能确定

(42) 有下列程序, 其中函数 f()的功能是将多个字符串按字典顺序排序 ( )。

```

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

```

程序运行后的输出结果是 ( )。

- A) 2                      B) 3                      C) 6                      D) 4

(43) 有下列程序:

```

#include <string.h>
void f(char *s,char *t)
{ char k;
 k=*s; *s=*t; *t=k;
 s+ +; t- -;
 if(*s) f(s,t);
}
main()
{ char str[10]="abcdefg",*p;
 p=str+strlen(str)/2+1;
 f(p,p-2);
}

```



```

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

```

程序运行后的输出结果是 ( )。

- A) abcdefg            B) gfedcba            C) gbcdefa            D) abedcfg

(44) 有下列程序:

```

float f1(float n)
{ return n*n; }
float f2(float n)
{ return 2*n; }
main()
{ float (*p1)(float),(*p2)(float),(*t)(float),y1,y2;
 p1=f1; p2=f2;
 y1=p2(p1(2.0));
 t=p1; p1=p2; p2=t;
 y2=p2(p1(2.0));
 printf("%3.0f, %3.0f\n",y1,y2);
}

```

程序运行后的输出结果是 ( )。

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

(45) 有下列程序:

```

int a=2;
int f(int n)
{ static int a=3;
 int t=0;
 if(n%2){ static int a=4; t+=a++;}
 else{ static int a=5; t+=a++;}
 return t+a++;
}
main()
{ int s=a,i;
 for(i=0;i<3;i++) s+=f(i);
 printf("%d\n",s);
}

```

程序运行后的输出结果是 ( )。

- A) 26            B) 28            C) 29            D) 24

(46) 有下列程序:

```

#include <string.h>
struct STU
{ int num;
 float TotalScore;
};
void f(structSTU p)
{ struct STU s[2]={ {20044,550},{20045,537}};
 p.num=s[1].num; p.TotalScore=s[1].TotalScore;
}

```

```

}
main()
{ struct STU s[2]={{20041,703},{20042,580}};
 f(s[0]);
 printf("%d %3.0f\n",s[0]. num,s[0]. TotalScore);
}

```

程序运行后的输出结果是 ( )。

- A) 20045 537      B) 20044 550      C) 20042 580      D) 20041 703

(47) 有下列程序:

```

#include <string.h>
struct STU
{ char name[10];
 int num;
};
void f(char *name,int num)
{ struct STU s[2]={"SunDan",20044},{"Penghua",20045}};
 num=s[0].num;
 strcpy(name,s[0].name);
}
main()
{ struct STU s[2]={{ "YangSan",2004},{ "LiSiGuo",20042}},*p;
 p=&s[1]; f(p->name,p->num);
 printf("%s %d\n", p->name,p->num);
}

```

程序运行后的输出结果是 ( )。

- A) SunDan 20042      B) SunDan 20044  
C) LiSiGuo 20042      D) YangSan 20041

(48) 有下列程序:

```

struct STU
{ char name[10]; int num; float TotalScore; };
void f(struct STU *p)
{ struct STU s[2]={"SunDan",20044,550}, {"Penghua",20045,537}},
 *q=s;+ +p; + +q; *p=*q;
}
main()
{ struct STU s[3]={{ "YangSan",20041,703},{ "LiSiGuo",20042,580}}};
 f(s);
 printf("%s%d%3.0f\n",s[1].name,s[1].num,s[1].TotalScore);
}

```

程序运行后的输出结果是 ( )。

- A) SunDan 20044 550      B) Penghua 20045 537  
C) LiSiGuo 20042 580      D) SunDan 20041 703

(49) 下列程序的功能是进行位运算:

```
main()
```

```

{ unsigned char a,b;
 a=7^3; b=~4 & 3;
 printf("%d%d\n",a,b);
}

```

程序运行后的输出结果是 ( )。

- A) 4 3                      B) 7 3                      C) 7 0                      D) 4 0

(50) 有下列程序:

```

#include <stdio.h>
main()
{ FILE *fp; int i,k,n;
 fp=fopen("data.dat","w+");
 for(i=1;i<6;i+ +)
 { fprintf(fp, "%d ",i);
 if(i%3==0) fprintf(fp,"\n");
 }
 rewind(fp);
 fscanf(fp,"%d%d",&k,&n); printf("%d%d\n",k,n);
 fclose(fp);
}

```

程序运行后的输出结果是 ( )。

- A) 0 0                      B) 123 45                      C) 1 4                      D) 1 2

## 二、填空题(每空 2 分, 共 40 分)

- (1) 数据管理技术发展过程经过人工管理、文件系统和数据库系统 3 个阶段, 其中数据独立性最高的阶段是 【1】。
- (2) 算法复杂度主要包括时间复杂度和 【2】 复杂度。
- (3) 在进行模块测试时, 要为每个被测试的模块另外设计两类模块: 驱动模块和承接模块(桩模块)。其中 【3】 的作用是将测试数据传送给被测试的模块, 并显示被测试模块所产生的结果。
- (4) 一棵二叉树第六层(根结点为第一层)的结点数最多为 【4】 个。
- (5) 数据结构分为逻辑结构和存储结构, 循环队列属于 【5】 结构。
- (6) 下列程序运行后的输出结果是 【6】。

```

main()
{ int x=0210; printf("%X\n",x);
}

```

- (7) 下列程序运行后的输出结果 【7】。

```

main()
{ int a=1,b=2,c=3;
 if(c=a) printf("%d\n",c);
 else printf("%d\n",b);
}

```

- (8) 已有定义: double \*p;, 请写出完整的语句, 利用 malloc 函数使 p 指向一个双精度型的动态存储单元 【8】。
- (9) 下列程序运行后的输出结果是 【9】。

```

main()
{ char c; int n=100;
 float f=10; double x;
 x=f* =n/ =(c=50);
 printf("%d %f\n",n,x);
}

```

(10) 下列程序的功能是计算：s=1+12+123+1234+12345。请填空。

```

main()
{ int t=0,s=0,i;
 for(i=1;i< =5;i+ +)
 { t=i+ 【10】; s=s+t; }
 printf("s=%d\n",s);
}

```

(11) 已知字母 A 的 ASCII 码为 65，下列程序运行后的输出结果是 **【11】**。

```

main()
{ char a,b;
 a='A'+'5'-'3'; b=a+'6'-'2';
 printf("%d %c\n",a,b);
}

```

(12) 有下列程序：

```

int sub(int n) { return(n/10+n%10);}
main()
{ int x,y;
 scanf("%d",&x);
 y=sub(sub(sub(x)));
 printf("%d\n",y);
}

```

若运行时输入:1234<CR>，程序的输出结果是 **【12】**。

(13) 下列函数 sstrcat()的功能是实现字符串的连接，即将 t 所指字符串复制到 s 所指字符串的尾部。例如：s 所指字符串为 abcd，t 所指字符串为 efgh，函数调用后 s 所指字符串为 abcdefgh。请填空。

```

#include <string.h>
void sstrcat(char *s,char *t)
{ int n;
 n=strlen(s);
 while(*(s+n)= 【13】){s+ +,t+ +;}
}

```

(14) 下列程序运行后的输出结果是 **【14】**

```

#include <string.h>
char *ss(char *s)
{ char *p,t;
 p=s+1;t=*s;
 while(*p){*(p-1)=*p;p+ +;}
 *(p-1)=t;
}

```

```

 return s;
 }
main()
{ char *p,str[10]="abcdefgh";
 p=ss(str);
 printf("%s\n",p);
}

```

(15) 下列程序运行后的输出结果是 **【15】**。

```

int f(int a[],int n)
{ if(n>=1)return f(a,n-1)+a[n-1];
 else return 0;
}
main()
{ int aa[5]={1,2,3,4,5},s;
 s=f(aa,5); printf("%d\n",s);
}

```

(16) 下列程序运行后的输出结果是 **【16】**。

```

struct NODE
{ int num; struct NODE *next;
};
main()
{ struct NODE s[3]={ {1, '\0'}, {2, '\0'}, {3, '\0'} },*p,*q,*r;
 int sum=0;
 s[0].next=s+1; s[1].next=s+2; s[2].next=s;
 p=s; q=p->next; r=q->next;
 sum+=q->next->num; sum+=r->next->next->num;
 printf("%d\n",sum);
}

```

(17) 下列程序的功能是输出如下形式的方阵：

```

13 14 15 16
9 10 11 12
5 6 7 8
1 2 3 4

```

请填写。

```

main()
{ int i,j,x;
 for(j=4;j【17】;j--)
 { for(i=1;i<=4;i++)
 { x=(j-1)*4+【18】;
 printf("%4d",x);
 }
 printf("\n");
 }
}

```

- (18) 下列函数 rotate 的功能是：将 a 所指 N 行 N 列的二维数组中的最后一行放到 b 所指二维数组的第 0 列中，把 a 所指二维数组中的第 0 行放到 b 所指二维数组的最后一列中，b 所指二维数组中其他数据不变。

```
#define N 4
void rotade(int a[][N],int b[][N])
{ int i,j;
 for(i=0;i<N;i++)
 { b[i][N-1]= 【19】; 【20】 =a[N-1][i]; }
}
```