

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 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)

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

(1) 数据的存储结构是指 D

- A) 存储在外存中的数据
- B) 数据所占的存储空间量
- C) 数据在计算机中的顺序存储方式
- D) 数据的逻辑结构中计算机中的表示

(2) 下列关于栈的描述中错误的是 B

- A) 栈是先进后出的线性表
- B) 栈只能顺序存储
- C) 栈具有记忆作用
- D) 对栈的插入与删除操作中, 不需要改变栈底指针

(3) 对于长度为 n 的线性表, 在最坏情况下, 下列各排序法所对应的比较次数中正确的是 D

- A) 冒泡排序为 $n/2$
- B) 冒泡排序为 n
- C) 快速排序为 n
- D) 快速排序为 $n(n-1)/2$

(4) 对长度为 n 的线性表进行顺序查找, 在最坏情况下所需要的比较次数为 C

- A) $\log_2 n$
- B) $n/2$
- C) n
- D) $n+1$

(5) 下列对于线性链表的描述中正确的是 A

- A) 存储空间不一定是连续, 且各元素的存储顺序是任意的
- B) 存储空间不一定是连续, 且前件元素一定存储在后件元素的前面
- C) 存储空间必须连续, 且前件元素一定存储在后件元素的前面
- D) 存储空间必须连续, 且各元素的存储顺序是任意的

(6) 下列对于软件的描述中正确的是 C

- A) 软件测试的目的是证明程序是否正确
- B) 软件测试的目的是使程序运行结果正确
- C) 软件测试的目的是尽可能多地发现程序中的错误
- D) 软件测试的目的是使程序符合结构化原则

(7) 为了使模块尽可能独立, 要求 B

- A) 模块的内聚程序要尽量高, 且各模块间的耦合程序要尽量强
- B) 模块的内聚程序要尽量高, 且各模块间的耦合程序要尽量弱
- C) 模块的内聚程序要尽量低, 且各模块间的耦合程序要尽量弱
- D) 模块的内聚程序要尽量低, 且各模块间的耦合程序要尽量强

(8) 下列描述中正确的是 D

- A) 程序就是软件

- B) 软件开发不受计算机系统的限制
- C) 软件既是逻辑实体, 又是物理实体
- D) 软件是程序、数据与相关文档的集合

(9) 数据独立性是数据库技术的重要特点之一。所谓数据独立性是指 D

- A) 数据与程序独立存放
- B) 不同的数据被存放在不同的文件中
- C) 不同的数据只能被对应的应用程序所使用
- D) 以上三种说法都不对

(10) 用树形结构表示实体之间联系的模型是 C

- A) 关系模型
- B) 网状模型
- C) 层次模型
- D) 以上三个都是

(11) 算法具有五个特性, 以下选项中不属于算法特性的是 B

- A) 有穷性
- B) 简洁性
- C) 可行性
- D) 确定性

(12) 以下选项中可作为 C 语言合法常量的是 A

- A) -80.
- B) -080
- C) -8e1.0
- D) -80.0e

(13) 以下叙述中正确的是 C

- A) 用 C 程序实现的算法必须要有输入和输出操作
- B) 用 C 程序实现的算法可以没有输出但必须要有输入
- C) 用 C 程序实现的算法可以没有输入但必须要有输出
- D) 用 C 程序实现的算法可以既没有输入也没有输出

14. 以下不能定义为用户标识符的是 D

- A) Main
- B) _0
- C) _int
- D) sizeof

(15) 以下选项中, 不能作为合法常量的是 B

- A) 1.234e04
- B) 1.234e0.4
- C) 1.234e+4
- D) 1.234e0

(16) 数字字符 0 的 ASCII 值为 48, 若有以下程序

```
main()
{
    char a='1', b='2';
    printf("%c,", b++);
    printf("%d\n", b-a);
}
```

程序运行后的输出结果是 C

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

(17)有以下程序

```
main()
{
    int m=12,n=34;
    printf("%d%d",m++,++n);
    printf("%d%d\n",n++,++m);
}
```

程序运行后的输出结果是 A

- A) 12353514
- B) 12353513
- C) 12343514
- D) 12343513

(18)有定义语句:int b;char c[10];,则正确的输入语句是 B

- A) scanf("%d%s",&b,&c);
- B) scanf("%d%s",&b,c);
- C) scanf("%d%s",b,c);
- D) scanf("%d%s",b,&c);

(19)有以下程序

```
main()
{
    int m,n,p;
    scanf("m=%dn=%dp=%d",&m,&n,&p);
    printf("%d%d%d\n",m,n,p);
}
```

若想从键盘上输入数据,使变量 m 中的值为 123,n 中的值为 456,p 中的值为 789,则正确的输入是 A

- A) m=123n=456p=789
- B) m=123 n=456 p=789
- C) m=123,n=456,p=789
- D) 123 456 789

(20)有以下程序

```
main()
{
    int a,b,d=25;
    a=d/10%9;
    b=a&&(-1);
    printf("%d,%d\n",a,b);
}
```

程序运行后的输出结果是 B

- A) 6,1
- B) 2,1
- C) 6,0

D) 2,0

(21)有以下程序

```
main()
{
    int i=1,j=2,k=3;
    if(i++==1&&(++j==3||k++==3))
        printf("%d %d %d\n",i,j,k);
}
```

程序运行后的输出结果是 D

A) 1 2 3

B) 2 3 4

C) 2 2 3

D) 2 3 3

(22)若整型变量 a、b、c、d 中的值依次为：1、4、3、2。则条件表达式 a<b?a:c<d?c:d 的值 A

A)1 B)2 c)3 D)4

(23)有以下程序

```
main()
{
    int p[8]={11,12,13,14,15,16,17,18}, i=0,j=0;
    while(i++<7) if(p[i]%2) j+=p[i];
    printf("%d\n",j);
}
```

程序运行后的输出结果是 B

A) 42

B) 45

C) 56

D) 60

(24) 有以下程序

```
main()
{
    char a[7]="a00a00";int i,j;
    i=sizeof(a); j=strlen(a);
    printf("%d %d\n",i,j);
}
```

程序运行后的输出结果是 C

A) 2 2

B) 7 6

C) 7 2

D) 6 2

(25)以下能正确定义一维数组的选项是 B

A) int a[5]={0,1,2,3,4,5};

B) char a[]={0,1,2,3,4,5};

C) char a={'A','B','C'};

D) int a[5]="0123";

(26)有以下程序

```
int fl(int x,int y){return x>y?x:y;}
```

```
int f2(int x,int y){return x>y?y:x;}
main()
{   int a=4,b=3,c=5,d=2,e,f,g;
    e=f2(f1(a,b),f1(c,d));
    f=f1(f2(a,b),f2(c,d));
    g=a+b+c+d-e-f;
    printf("%d,%d%d\n",e,f,g);
}
```

程序运行后的输出结果是

A

A) 4, 3, 7

B) 3, 4, 7

C) 5, 2, 7

D) 2, 5, 7

(27) 已有定义: char a[]="xyz",b[]={'x','y','z'};,以下叙述中正确的是 C

A) 数组 a 和 b 的长度相同

B) a 数组长度小于 b 数组长度

C) a 数组长度大于 b 数组长度

D) 上述说法都不对

(28) 有以下程序

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

```
{   int t;
```

```
    t=*x;*x=*y;*y=t;
```

```
}
```

```
main()
```

```
{   int a[8]={1,2,3,4,5,6,7,8},i,*p,*q;
```

```
    p=a;q=&a[7];
```

```
    while(p<q)
```

```
    {
```

```
        f(p,q);
```

```
        p++;
```

```
        q--;
```

```
    }
```

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

```
        printf("%d,",a[i]);
```

```
}
```

程序运行后的输出结果是

D

A) 8, 2, 3, 4, 5, 6, 7, 1,

B) 5, 6, 7, 8, 1, 2, 3, 4,

C) 1, 2, 3, 4, 5, 6, 7, 8,

D) 8, 7, 6, 5, 4, 3, 2, 1,

(29) 有以下程序

```
main()
```

```
{
```

```
int a[3][3],*p,i;
p=&a[0][0];
for(i=0;i<9;i++)
    p[i]=i;
for(i=0;i<3;i++)
    printf("%d ",a[1][i]);
}
```

程序运行后的输出结果是 D

- A) 0 1 2
- B) 1 2 3
- C) 2 3 4
- D) 3 4 5

(30)以下叙述中错误的是 C

- A) 对于 double 类型数组，不可以直接用数组名对数组进行整体输入或输出
- B) 数组名代表的是数组所占存储区的首地址，其值不可改变
- C) 当程序执行中，数组元素的下标超出所定义的下标范围时，系统将给出“下标越界”的出错信息
- D) 可以通过赋初值的方式确定数组元素的个数

(31) 有以下程序

```
#define N 20
fun(int a[],int n,int m)
{ int i,j;
  for(i=m;i>=n;i--)
    a[i+1]=a[i];
}
main()
{
  int i,a[N]={1,2,3,4,5,6,7,8,9,10};
  fun(a,2,9);
  for(i=0;i<5;i++)
    printf("%d",a[i]);
}
```

程序运行后的输出结果是 C

- A) 10234
- B) 12344
- C) 12334
- D) 12234

(32) 有以下程序

```
main()
{ int a[3][2]={0},(*ptr)[2],i,j;
  for(i=0;i<2;i++)
  { ptr=a+i;
    scanf("%d",ptr);
    ptr++;
  }
}
```

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

若运行时输入:1 2 3<回车>,则输出结果为

B

A) 产生错误信息

B) 1 0

2 0

0 0

C) 1 2

3 0

0 0

D) 1 0

2 0

3 0

(33)有以下程序

```
prt(int *m,int n)
{
    int i;
    for(i=0;i<n;i++)
        m[i]++;
}
main()
{
    int a[]={1,2,3,4,5},i;
    prt(a,5);
    for(i=0;i<5;i++)
        printf("%d,",a[i]);
}
```

程序运行后的输出结果是: B

A) 1, 2, 3, 4, 5,

B) 2, 3, 4, 5, 6,

C) 3, 4, 5, 6, 7,

D) 2, 3, 4, 5, 1,

(34) 有以下程序

```
main()
{
    int a[]={1,2,3,4,5,6,7,8,9,0},*p;
    for(p=a;p<a+10;p++)
        printf("%d,",*p);
}
```

程序运行后的输出结果是

A

- A) 1, 2, 3, 4, 5, 6, 7, 8, 9, 0,
- B) 2, 3, 4, 5, 6, 7, 8, 9, 10, 1,
- C) 0, 1, 2, 3, 4, 5, 6, 7, 8, 9,
- D) 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,

(35) 有以下程序

```
#define P 3
#define F(int x) { return (P*x*x);}
main()
{
    printf("%d\n",F(3+5));
}
```

程序运行后的输出结果是

D

- A) 192
- B) 29
- C) 25
- D) 编译出错

(36) 有以下程序

```
main()
{
    int c=35; printf("%d\n",c&c);
}
```

程序运行后的输出结果是

C

- A) 0
- B) 70
- C) 35
- D) 1

(37) 以下叙述中正确的是

D

- A) 预处理命令行必须位于源文件的开头
- B) 在源文件的一行上可以有多条预处理命令
- C) 宏名必须用大写字母表示
- D) 宏替换不占用程序的运行时间

(38) 若有以下说明和定义

```
union dt
{
    int a; char b; double c;
}data;
```

以下叙述中错误的是

C

- A) data 的每个成员起始地址都相同
- B) 变量 data 所占内存字节数与成员 c 所占字节数相等
- C) 程序段: data.a=5;printf("%f\n",data.c);输出结果为 5.000000
- D) data 可以作为函数的实参

(39) 以下语句或语句组中, 能正确进行字符串赋值的是

D

- A) char *sp; *sp="right!";
- B) char s[10]; s="right!";

C) char s[10]; *s="right!";

D) char *sp="right!";

(40)设有如下说明

typedef struct ST

```
{  
    long a;  
    int b;  
    char c[2];
```

```
} NEW;
```

则下面叙述中正确的是 C

A) 以上的说明形式非法

B) ST 是一个结构体类型

C) NEW 是一个结构体类型

D) NEW 是一个结构体变量

(41) 有以下程序

main()

```
{  
    int a=1,b;  
    for(b=1;b<=10;b++)  
    {  
        if(a>=8) break;  
        if(a%2==1) { a+=5; continue;}  
        a-=3;  
    }  
    printf("%d\n",b);  
}
```

程序运行后的输出结果是 B

A) 3

B) 4

C) 5

D) 6

(42) 有以下程序

main()

```
{  
    char s[]="159",*p;  
    p=s;  
    printf("%c",*p++);  
    printf("%c",*p++);  
}
```

程序运行后的输出结果是 A

A) 15

B) 16

C) 12

D) 59

(43) 有以下函数

```
fun(char *a,char *b)
{
    while((*a!='\0')&&(*b!='\0')&&(*a==*b))
    {   a++;   b++;}
    return (*a-*b);
}
```

该函数的功能是 **D**

- A) 计算 a 和 b 所指字符串的长度之差
- B) 将 b 所指字符串复制到 a 所指字符串中
- C) 将 b 所指字符串连接到 a 所指字符串后面
- D) 比较 a 和 b 所指字符串的大小

(44) 有以下程序

```
main()
{
    int num[4][4]={ {1,2,3,4},{5,6,7,8},{9,10,11,12},{13,14,15,16}},i,j;
    for(i=0;i<4;i++)
    {
        for(j=1;j<=i;j++)   printf("%c",' ');
        for(j=   j<4;j++)   printf("%4d",num[i][j]);
        printf("\n");
    }
}
```

若要按以下形式输出数组右上半三角

```
1 2 3 4
   6 7 8
      11 12
         16
```

则在程序下划线处应填入的是 **B**

- A) i-1
- B) i
- C) i+1
- D) 4-i

(45) 有以下程序

```
point(char *p)
{
    p+=3;
}
main()
{   char b[4]={'a','b','c','d'},*p=b;
    point(p);   printf("%c\n",*p);
}
```

程序运行后的输出结果是 **A**

- A) a
- B) b
- C) c
- D) d

(46)程序中若有如下的说明和定义语句

```
char fun(char *);  
main()  
{  
    char *s="one",a[5]={0},(*f1)=fun,ch;  
    .....  
}
```

以下选项中对函数 fun 的正确调用语句是 A

- A) (*f1) (a);
- B) *f1(*s);
- C) fun(&a);
- D) ch=*f1(s)

(47)有以下结构体说明和变量定义，如图所示，指针 pqr 分别指向此链表中三个连续结点。

```
struct node  
{  
    int data;  
    struct node *next;  
} *p,*q,*r;
```

现要将 q 所指结点从链表中删除，同时保持链表的连续,以下不能完成指定操作的语句是 D

- A) p->next=q->next;
- B) p->next=p->next->next;
- C) p->next=r;
- D) p=q->next;

(48)以下对结构体类型变量 td 的定义中，错误的是 C

- A) typedef struct aa
 {
 int n;
 float m;
 }AA;
 AA td;
- B) struct aa
 {
 int n;
 float m;
 } td;
 struct aa td;
- C) struct
 {

```
int n;
float m;
}aa;
struct aa td;
D) struct
{
int n;
float m;
}td;
```

(49)以下与函数 fseek(fp,0L,SEEK_SET)有相同作用的是 D

- A) feof(fp)
- B) ftell(fp)
- C) fgetc(fp)
- D) rewind(fp)

(50)有以下程序

```
#include "stdio.h"
void WriteStr(char *fn,char *str)
{
FILE *fp;
fp=fopen(fn,"W");
fputs(str,fp);
fclose(fp);
}
main()
{
WriteStr("t1.dat","start");
WriteStr("t1.dat","end");
}
```

程序运行后,文件 t1.dat 中的内容是

B

- A) start
- B) end
- C) startend
- D) endrt

待续.....

2005 年 4 月 2 日 C 笔试试卷及答案,快来对答案(三)!!!!!!

填空题(每空 2 分,共 40 分)

- (1) 某二叉树中,度为 2 的结点有 18 个,则该二叉树中有 19 个叶子结点。
- (2) 在面向对象的方法中,类的实例称为 对象。
- (3) 诊断和改正程序中错误的工作通常称为程序调试。
- (4) 在关系数据库中,把数据表示成二维表,每一个二维表称为关系。
- (5) 问题处理方案的正确而完整的描述称为 算法。
- (6) 以下程序运行时若从键盘输入: 10 20 30<回车>。输出结果是 10 30 0。

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

```
main()
{   int i=0,j=0,k=0;
    scanf("%d%d%d",&i,&j,&k);
    printf("%d%d%d\n",i,j,k);
}
```

(7)以下程序运行后的输出结果是 81 .

```
#define S(x)  4*x*x+1
```

```
main()
{
    int i=6,j=8;
    printf("%d\n",S(i+j));
}
```

(8)以下程序运行后的输出结果是 4599

```
main()
{
    int a=3,b=4,c=5,t=99;
    if(b<a&& a<c)  t=a;a=c;c=t;
    if(a<c&& b<c)  t=b;b=a;a=t;
    printf("%d%d%d\n",a,b,c);
}
```

(9)以下程序运行后的输出结果是 10 20 0

```
main()
{
    int a,b,c;
    a=10;b=20;c=(a%b<1)|| (a/b>1);
    printf("%d %d %d\n",a,b,c);
}
```

(10)以下程序运行后的输出结果是 0918273645

```
main()
{
    char c1,c2;
    for(c1='0',c2='9';c1<c2;c1++,c2--)
        printf("%c%c",c1,c2);
    printf("\n");
}
```

(11)已知字符 A 的 ASCII 代码值为 65,以下程序运行时若从键盘输入:B33<回车>,输出结果是 1B

```
#include "stdio.h"
main()
{
    char a,b;
    a=getchar(); scanf("%d",&b);
    a=a-'A'+'0';
    b=b*2;
    printf("%c %c\n",a,b);
}
```

```
}
```

(12)以下程序中,fun 函数的功能是求 3 行 4 列二维数组每行元素中的最大值.请填空. br[i]

```
void fun(int, int, int (*)[4],int *)
```

```
main()
```

```
{
```

```
int a[3][4]={ { 12,41,36,28},{ 19,33,15,27},{ 3,27,19,1 } },b[3],i;
```

```
fun(3,4,a,b);
```

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

```
printf("%4d",b[i]);
```

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

```
}
```

```
void fun(int m,int n,int ar[][4],int *br)
```

```
{
```

```
int i,j,x;
```

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

```
{ x=ar[i][0];
```

```
for(j=0;j<n;j++)
```

```
if(x<ar[i][j]) x=ar[i][j];
```

```
【12】 =x;
```

```
}
```

(13)以下程序运行后的输出结果是 4 3 3 4

```
void swap(int x,int y)
```

```
{ int t;
```

```
t=x;x=y;y=t;
```

```
printf("%d %d ",x,y);
```

```
}
```

```
main()
```

```
{ int a=3,b=4;
```

```
swap(a,b);
```

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

```
}
```

(14)以下程序运行后的输出结果是 abcfg

```
#include "string.h"
```

```
void fun(char *s,int p,int k)
```

```
{ int i;
```

```
for(i=p;i<k-1;i++)
```

```
s[i]=s[i+2];
```

```
}
```

```
main()
```

```
{ char s[]="abcdefg";
```

```
fun(s,3,strlen(s));
```

```
puts(s);
```

```
}
```

(15)以下程序运行后的输出结果是 abcbcc

```
#include "string.h"
main()
{
    char ch[]="abc",x[3][4]; int i;
    for(i=0;i<3;i++) strcpy(x[i],ch);
    for(i=0;i<3;i++) printf("%s",&x[i][i]);
    printf("\n");
}
```

(16)以下程序运行后的输出结果是 0 10 1 11 2 12

```
fun(int a)
{
    int b=0;static int c=3;
    b++; c++;
    return (a+b+c);
}
```

```
main()
{
    int i,a=5;
    for(i=0;i<3;i++)
        printf("%d %d ",i,fun(a));
    printf("\n");
}
```

(17)以下程序运行后的输出结果是 13431

```
struct NODE
{ int k;
  struct NODE *link;
};
main()
{
    struct NODE m[5],*p=m,*q=m+4;
    int i=0;
    while(p!=q)
    { p->k=++i; p++;
      q->k=i++; q--;
    }
    q->k=i;
    for(i=0;i<5;i++)
        printf("%d",m[i].k);
    printf("\n");
}
```

(18)以下程序中函数 `huiwen` 的功能是检查一个字符串是否是回文,当字符串是回文时,函数返回字符串:yes

串:yes,否则函数返回字符串:no!,并在主函数中输出.所谓回文即正向与反向的拼写都一样,例如:adgda.请填空.

```
#include "string.h"
char *huiwen(char *str)
{
    char *p1,*p2; int i,t=0;
    p1=str; p2= ①
    for(i=0;i<=strlen(str)/2;i++)
        if(*p1++!=*p2--)
            { t=1; break; }
    if( ②) return ("yes!");
    else return ("no!");
}
main()
{
    char str[50];
    printf("Input:"); scanf("%s",str);
    printf("%s\n", ③ );
}
①str+strlen(str)-1
②t==0 或 !t
③huiwen(str)
```