

2015 年计算机二级《C 语言》最新章节练习题(3)

一、选择题

1、 有以下程序

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

程序运行后输出结果是

A. ABE

B. BCD

C. BDI

D. BCF

2、 有以下程序

```
#include <stdio.h>
int f(int k)
{ static int s=1;
  *s=k;
  *s=s;
}
main()
{ int m=3;
  f(&m);
  printf(" %d, %d\n", m, k);
}
```

程序运行后的输出结果是

A. 3, 5

B. 5, 5

C. 3, 3

D. 5, 3

3、有以下程序

```
#include <stdio.h>
main( )
{ char c;
  c = getchar();
  do {
    putchar(c++);
  } while((c = getchar()) != '#');
}
```

程序运行时从第一列开始输入：abcdefg##<回车>，则输出结果是

A. abcdefg

B. bcddgh

C. abcdefg#

D. bcddgh\$

4、有以下程序

```
#include <stdio.h>
#define N 4
void fun(int a[][N])
{
  int i;
  for(i=0; i<N; i++)
    a[i][i] += a[N-1][N-1-i];
}
main()
{ int x[N][N] = {
  {1, 2, 3, 4},
  {5, 6, 7, 8},
  {9, 10, 11, 12},
  {13, 14, 15, 16}
}, i;
  fun(x);
  for(i=0; i<N; i++) printf("%d, ", x[i][i]);
  printf("\n");
}
```

程序运行后的输出结果是

A. 17, 6, 11, 16,

B. 1, 6, 11, 16,

C. 4, 7, 10, 13,

D. 5, 13, 21, 29,

5、 以下选项中，能正确进行字符串赋值的是

```
A) char *s = "ABCDE" ;  
B) char s[5] = { 'A', 'B', 'C', 'D', 'E' };  
C) char s[4][5] = { "ABCDE" };  
D) char *s ; gets( s );
```

6、 关于

do 循环体 while(条件表达式);

以下叙述中正确的是

A. 条件表达式的执行次数总是比循环体的执行次数多一次

B. 循环体的执行次数总是比条件表达式的执行次数多一次

C. 条件表达式的执行次数与循环体的执行次数一样

D. 条件表达式的执行次数与循环体的执行次数无关

7、 若有以下程序

```
#include <stdio.h>  
char f(char x)  
{  
    return x * 10 + '0';  
}  
main()  
{  
    char a; int b=0;  
    for ( a=0; a<5; a++ )  
        b = f(a) ;    putchar(B) ;  
}
```

则程序的输出结果是

A. 014916

B. 0123 4

C. 01496

D. 00000

8、与数学表达式 $x \geq y \geq z$ 对应的 c 语言表达式是

A. $(x >= y) \parallel (y >= x)$

B. $(x >= y >= z)$

C. $(x >= y) ! (y >= z)$

D. $(x >= Y) \&\& (y >= z)$

9、若有以下程序

```
#include <stdio.h>
void f(int x)
{
    if (x >= 10)
    {
        printf("%d - ", x%10);    f(x/10);
    }
    else
        printf("end\n");
}
main()
{
    x = 123456;
```

则程序的输出结果是

A. 6-5-4-3-2-1-

B. 6-5-4-3-2-1

C. 1-2-3-4-5-6

D. 1-2-3-4-5-6

10、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. bc2yz

C. a2yz

D. 12yz

11、 有以下程序

```

#include <stdio.h>
int f1(int p)
{
    int t=5;
    d = p++;
    printf("%d ", d);
}
main()
{
    int i=3;
    fun( i );
    i = i++;
    printf("%d\n", i);
}

```

程序的输出结果是

A. 8 4

B. 9 6

C. 9 4

D. 8 5

12、 若有以下程序

```

#include <stdio.h>
main()
{
    FILE *fp;
    int i, a[6] = {1,2,3,4,5,6}, k;
    fp = fopen("data.dat", "w+");
    for (i=0; i<6; i++)
        fseek(fp, 0L, 0);
        fprintf(fp, "%d\n", a[i]);
    rewind(fp);
    fscanf(fp, "%d", &k);
    fclose(fp);
    printf("%d\n", k);
}

```

则程序的输出结果是

A. 123456

B. 1

C. 6

D. 21

13、下面选项中关于编译预处理的叙述正确的是

A. 预处理命令行必须使用分号结尾

B. 凡是以#号开头的行，都被称为编译预处理命令行

C. 预处理命令行不能出现在程序的最后一行

D. 预处理命令行的作用域是到最近的函数结束处

14、有以下程序段

```

int *p;
p = _____ malloc( sizeof( int ) );

```

若要求使P指向一个 int 型的动态存储单元，在横线处应填入的是

A. (int*)

B. int

C. int*

D. (*int)

15、 设 P 是指针变量，语句 P=NULL; 等价于

A. P='0';

B. P="";

C. P=""

D. P=0;

16、 有以下程序

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

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

程序运行后的输出结果是

A. 34756

B. 5745

C. 5795

D. 35745

17、 以下叙述中正确的是

- A. 使用 typedef 说明新类型名时，其格式是：typedef 新类型名原类型名；
- B. 在程序中，允许用 typedef 来说明一种新的类型名
- C. 使用 typedef 说明新类型名时，后面不能加分号
- D. 在使用 typedef 改变原类型的名称后，只能使用新的类型名

18、 有以下程序

```
#include <stdio.h>
#define SQR(X) X * X
main()
{
    int a=10, k=2, m=1;
    a /= SQR(k+m)/SQR(k+m);
    printf( "%d\n", A) ;
}
```

程序的输出结果是

- A. 0
- B. 1
- C. 9
- D. 10

19、 以下叙述中正确的是

- A. 语句 `int a[4][3]={1, 2}, {4, 5};` 是错误的初始化形式
- B. 语句 `int a[4][3]={1, 2, 4, 5};` 是错误的初始化形式
- C. 语句 `int a[4][3]={1, 2, 4, 5};` 是错误的初始化形式
- D. 在“二维数组”中，可以把二维数组看成是一个具有行和列的表格或矩阵

20、 有以下程序


```

#include <stdio.h>
main()
{
    char *s="12134";
    int k=0, a=0;
    while( s[ k + 1 ] )
    {
        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=5 a=15

B. k=4 a=12

C. k=6 a=11

D. k=5 a=14

21. 若有以下程序

```

#include <stdio.h>
void sp(int *A)
{
    int b=2;
    a=&b; *a = *a * 2; printf( "%d", *A );
}
main()
{
    int k=3, *p=&k;
    sp(p); printf( "%d,%d\n", k, *p );
}

```

则程序的输出结果是

A. 4, 3, 3

B. 4, 3, 4

C. 6, 3, 6

D. 6, 6, 6

22、 以下涉及字符串数组、字符指针的程序段，没有编译错误的是

```
A) char * str, name[5];  
   name = "Hello";  
B) char * str, name[6];  
   str = "C/C++";  
C) char str1[7] = "prog. c", str2[8];  
   line = "////////";  
D) char line[];  
   str2 = str1;
```

23、 有如下程序

```
#include <stdio.h>  
main()  
{  
    int a = 0, b = 1;  
    if (a++ && b++)  
        printf("TRUE");  
    else  
        printf("FALSE");  
    printf("_OUT: a=%d, b=%d\n", a, b);  
}
```

程序运行后的输出结果是

A. FALSE OUT: a=1, b=1

B. FALSE_OUT: a=0, b=2

C. TRUE_OUT: a=1, b=2

D. TRUE_OUT: a=0, b=11

24、 以下叙述中正确的是

A. 一条语句只能定义一个数组

B. 每个数组包含一组具有同一类型的变量，这些变量在内存中占有连续的存储单元

C. 数组说明符的一对方括号中只能使用整型常量，而不能使用表达式

D. 在引用数组元素时，下标表达式可以使用浮点数

25、 以下叙述中正确的是

A. 基类型不同的指针变量可以相互混用

B. 函数的类型不能是指针类型

C. 函数的形参类型不能是指针类型

D. 设有指针变量为 `double*p`，则 `P+1` 将指针 `P` 移动 8 个字节

26、 有如下的说明和定义

```
struct t {
    int a;
    char *s;
} *p;
x.a = 1;
x.s = "hello";
```

则以下叙述中正确的是

A. `(P++)->a` 与 `P++->a` 都是合语法的表达式，但二者不等价

B. 语句 `++P->a` 的效果是使 `P` 增 1

C. 语句 `++P->a` 的效果是使成员 `a` 增 1

D. 语句 `P->s++` 等价于 `(*P)->s++`;

27、 若 `fp` 已定义为指向某文件的指针，且没有读到该文件的末尾，则 C 语言函数 `feof(fp)` 的函数返回值是

A. EOF

B. 非 0

C. -1

D. 0

28、 有以下程序

```
#include <stdio.h>
main()
{
    int a[ ] = { 2,4,6,8 }, *p=a, i;
    for( i=0; i<4; i++ )
        a[i] = *p++;
    printf( "%d\n", a[2] );
}
```

程序的输出结果是

A. 2

B. 8

C. 4

D. 6

30、 若有以下程序

```
#include <stdio.h>
main()
{
    int a = -2, b = 0;
    while( a++ )
        printf( "%d, ", b );
}
```

则程序的输出结果是

A. 1, 2

B. 0, 2

C. 1, 3

D. 2, 3