

2015 年计算机二级《C 语言》最新章节练习题(6)

一、选择题

1 下面关于“EOF”的叙述，正确的是

- A. EOF 的值等于 0
- B. EOF 是在库函数文件中定义的符号常量
- C. 文本文件和二进制文件都可以用 EOF 作为文件结束标志
- D. 对于文本文件，fgetc 函数读入最后一个字符时，返回值是 EOF

参考答案：B

参考解析：在 c 语言中，或更精确地说成 c 标准函数库中表示文件结束符(end of file)。在 while 循环中以 EOF 作为文件结束标志。这种以 EOF 作为文件结束标志的文件，必须是文本文件。在文本文件中，数据都是以字符的 ASCII 代码值的形式存放。我们知道，ASCII 代码值的范围是 0~255，不可能出现-1，因此可以用 EOF 作为文件结束标志。

2 有以下程序

```
#include <stdio.h>
char fun(char c)
{
    if ( c <= 'Z' && c >= 'A' )
        *c = 'A' - 'a';
    return *c;
}

main()
{
    char s[81], *p=s;
    gets(s);
    while( *p )
    {
        *p = fun( *p );
        putchar( *p );
        p++;
    }
    printf( "\n" );
}
```

若运行时从键盘上输入 OPEN THE DOOR<回车>，程序的输出结果是

A. OPEN THE DOORB. oPEN tHE dOORC. open the doorD. Open The Door

参考答案: C

参考解析: 字符串输入函数 gets 的功能是从标准输入设备键盘上输入一个字符串。首先使指针变量 P 指向字符串的首字母, while 循环语句中对字符串的每个字符进行 fun 函数操作。fun 函数的功能是, 将字符串中的大写字母变小写字母。将处理结果返回主函数, 主函数通过 putchar() 字符输出函数进行输出。因此 C 选项正确。

3 为了建立如图所示的存储结构(即每个结点含两个

```
*t=k; |
    return s;
|
main()
|
| int i=3, j=5, *p=&i, *q=&j, *r;
| r=f(p,q);
| printf( "%d,%d,%d,%d,%d\n", i, j,
| *p, *q, *r);
|
```

A. link next;B. struct link*next;C. link*next;D. struct link next;

参考答案: B

参考解析: 本题主要考查链表中结点的概念, 属于基础知识, 其中指针域应该是指向下个结点, 所以答案为 B。

4 若有以下程序

```
#include <stdio.h>
main()
{
    int n;
    for(n=0; n<4; n++)
    {
        switch(n)
        {
            default: s+=4;
            case 1: s+=1; break;
            case 2: s+=2; break;
            case 3: s+=3;
        }
    }
    printf( "%d\n", s);
}
```

则程序的输出结果是

A. 10B. 11C. 13D. 15

参考答案: B

参考解析：break 语句的作用是终止正在执行的 switch 流程，跳出 switch 结构或者强制终止当前循环，从当前执行的循环中跳出。题干中第一次循环 n 值为 0，执行 default 语句后的 s+=4，s 的值变为 4，执行 case1 语句后的 s+=1，s 的值变为 5，遇到 break 语句跳出 switch 语句，进入第二次循环。第二次循环时 n 的值为 1。执行 case1 后的 s+=1，s 的值变为 6。遇到 break 语句跳出 switch 语句，进入第三次循环。第三次循环时 n 的值为 2，执行 case2 后的 s+=2，s 的值变为 8，遇到 break 语句跳出 switch 语句，进入第四次循环。第四次循环时 n 的值为 3，执行 case3 后的 s+=3，s 的值变为 11。再判断循环条件为假，退出循环打印 s 的值 11。

5 有以下程序

```
int fun( char *s )
{
    char *p = s;
    while ( *p != '\0' ) p++;
    return ( p - s );
}

main()
{
    printf( "%d\n", fun( "goodbey!" ) );
}
```

程序的输出结果是

A. 0 B. 6 C. 7 D. 8

参考答案：B

参考解析：本题主要考查了函数定义及调用，以及 while 循环语句，执行 fun 函数后，首先是找到最后一个字符的地址，然后减去第一个地址得到的就是这个字符串的长度，所以本题答案为 B。

6 有以下程序

```
#include <stdio.h>
int fun( int n )
{
    int a;
```

```

    if( n == 1 ) return 1;
    a = n + fun( n-1 );
    return ( a );
}
main()
{ printf( "%d\n", fun( 5 ) ); }

```

程序的输出结果是

A. 9 B. 14 C. 10 D. 15

参考答案: D

参考解析: 考查函数的递归调用, 只需要一步一步递归调用 fun 函数即可, 执行 $\text{fun}(5) = 5 + (4 + (3 + (2 + \text{fun}(1)))) = 15$ 。所以答案为 D 选项。

7 以下叙述中正确的是

- A. 结构体数组名不能作为实参传给函数
- B. 结构体变量的地址不能作为实参传给函数
- C. 结构体中可以含有指向本结构体的指针成员
- D. 即使是同类型的结构体变量, 也不能进行整体赋值

参考答案: C

参考解析: 本题考查结构体变量作为函数参数, 结构体变量的地址可以作为参数传给函数, 结构体数组名就是一个地址, 可以传给函数, 如果是同类型的结构体变量, 可以整体赋值。

8 若有以下程序

```

#include <stdio.h>
int *f(int *s, int *t)
{ int k;
  if ( *s < *t ) k = *s; *s = *t;
}

```

```

    *t=k; |
        return s;
    |
    |
main( )
|
|   int  i=3, j=5, *p=&i, *q=&j, *r;
|   r=f(p,q);
|   printf( "%d,%d,%d,%d,%d\n", i, j,
|           *p, *q, *r);
|

```

则程序的输出结果是

A. 5, 3, 3, 5, 5 B. 3, 5, 5, 3, 5 C. 3, 5, 3, 5, 5 D. 5, 3, 5, 3, 5

参考答案: D

参考解析: 在主函数中分别给整型变量 i、j 和指针型变量 P、q 赋初值, 并声明指针变量 r; 调用 f 函数, 并将实参变量 P 和 q 的值传递给形参变量 s 和 t, 并且 f 函数是指针型函数, 即函数的返回值将是一个地址。在 f 函数中, 如果条件成立, 则将指针变量 s 和 t 的值互换, 并且将指针 S 的地址返回主函数。最后输出 i, j, *P, *q, *r 的值, 即 5, 3, 5, 3, 5。因此 D 选项正确。

9 以下叙述中正确的是

A. 在 c 语言中, 逻辑真值和假值分别对应 1 和 0

B. 关系运算符两边的运算对象可以是

C 语言中任意合法的表达式 C. 对于浮点变量 x 和 y, 表达式: x==y 是非法的, 会出编译错误

D. 分支结构是根据算术表达式的结果来判断流程走向的

参考答案: B

参考解析: A 选项中, 在 c 语言中, 逻辑真值对应非 0; C 选项中, 表达式: x==y 是合法的; D 选项中, 分支结构的流程走向是根据表达式的值, 并不仅仅是算数表达式的值。因此 B 选项正确。

10 有如下程序

```

#include <stdio.h>
struct pair
{
    int first, second;
};
struct pair get_min_max(int * array, int len)
{
    int i;
    struct pair res;
    res.first = array[0];
    res.second = array[0];
    for (i = 1; i < len; i++)
    {
        if (array[i] < res.first)
            res.first = array[i];
        if (array[i] > res.second)
            res.second = array[i];
    }
    return res;
}
main( )
{
    int array[6] = {1, 2, 3, 4};
    struct pair min_max = get_min_max(array, 6);

    printf("min = %d, max = %d\n", min_max.
first, min_max.second);
}

```

程序运行后的输出结果是

A. min=1, max=21 B. min=1, max=2 C. min=1, max=19 D. min=0, max=6

参考答案: A

参考解析: 本题首先在 main 函数定义一个数组 array, 然后定义了一个结构变量 min_max, min_max 使用函数 get_min_max 的返回值初始化, 函数 get_min_max 的功能就是将数组中的最小值和最大值取出来后分别赋值给结构变量 min_max 中的 first, second 两个整形变量, 数组长度为 6 个单位, 初始化 4 个数据, 剩下 2 个默认为 0, 因此答案为 A 选项。

11 有以下程序

```

#include <stdio.h>
fun( int a, int b )
{
    int static m=0, i=2;
    i = i + m + 1;
    m = i + a + b;
    return m;
}

main()
{
    int k=4, m=1, p;
    p = fun( k, m);
    printf( "%d", p);
    p = fun( k, m);
    printf( "%d\n", p);
}

```

程序运行后的输出结果是

A. 8, 17 B. 7, 16 C. 8, 8 D. 7, 17

参考答案: A

参考解析: 本题考查静态局部变量, 本题中先执行 fun(4, 1) 得到返回值为 8, 此时静态局部变量 m 为 8, i 为 2, 那么再次执行 fun(4, 1) 得到的值为 17, 所以选项 A 正确。

12 有以下程序

```

#include <stdio.h>
#define N 10
#define M N+1
#define NUM (M+1)*M/2
main()
{ printf( "%d\n", NUM ); }

```

程序运行后的输出结果是

A. 4 B. 8 C. 9 D. 6

参考答案: B

参考解析: 本题考查宏定义, 宏定义只是做个简单的替换, 执行 $NUM = (N+1+1) * N + 1 / 2 = 8$, 选项 B 正确。

13 有以下程序

```

#include <stdio.h>
int fun( char s[] )
{
    char *p = s;
    while( *p != 0 ) p++;
    return ( p-s );
}

main()
{
    printf( "%d\n", fun( "0ABCDEF" ) );
}

```

程序运行后的输出结果是

A. 1B. 6C. 7D. 0

参考答案: C

参考解析: 本题考查字符串作为函数参数以及 while 语句, 本题中传入字符串 0ABCDEF, 然后执行 fun 后, 就得到了该字符串中字符的个数, 不包括结尾符, 该字符串有 7 个字符, 所以答案为 C。

14 设有定义: `int a[10]={0, 1, 2, 3, 4, 5, 6, 7, 8, 9};` 若 $0 \leq i \leq 9$, 则对 a 数组元素的引用错误的是

A. `a[10]` B. `*a[i]` C. `P[i]` D. `[P-a]`

参考答案: A

参考解析: 长度为 n 的数组其各个元素的下标应该是从 0 到 n-1, 因此, 长度为 10 的数组 a, 第 10 个元素为 `a[9]`, 而不是 `a[10]`, 答案为 A 选项。

15 若有以下程序

```

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

main()
{
    int k=3;
    sp(&k); printf( "%d\n", k );
}

```

则程序的输出结果是

A. 6. 12, 12B. 6, 4, 3C. 6, 4, 6D. 9, 4, 9

参考答案: C

参考解析: 首先在主函数中给整型变量 k 赋值为 3, 调用 sp 函数, 将变量 k 的地址作为实参传递给形参指针变量 a; 在 sp 函数中, $*a = *a * 2$; 即结果为 6 由于指针变量 a 指向的地址 k, 所以 k 值此时也变成 6, 通过 printf() 函数输出。整型变量 b 的地址赋给指针变量 a, 再执行 $*a = *a * 2$;。即结果为 4, 通过 printf() 函数输出。返回主函数, 输出 k 的值 6。因此 C 选项正确。

16 有以下程序

```
#include <stdio.h>
void fun( char *p, int n )
{ char b[6] = "abcde"; int i;
  for( i = 0, p = b; i < n; i++ ) p[i] =
b[i];
}
main()
{ char a[6] = "ABCDE";
  fun(a, 5); printf("%s\n", a);
}
```

程序运行后的输出结果是

A. abcdeB. ABCDEcC. dcbcdD. EDCBA

参考答案: B

参考解析: 本题考查数组名作为函数参数, 执行 f 函数时, 传进去的 a 指针被重新指向了 b, 所以原本 a 数组的地址内容不变, 所以输出结果为 ABCDE, 答案为 B 选项。

17 有以下程序

```
#include <stdio.h>
int funa( int a, intB ) | return a + b; |
int funb( int a, intB ) | return a - b; |
int sss( int (*t)(), int x, int y ) |
return( (*t)(x,y) ); |
main()
{ int x;
  x = sss( funa, 9,3 ); x += sss( funb,8,3 );
  printf( "%d\n", x );
}
```

程序运行后的输出结果是

A. 24B. 23C. 17D. 22

参考答案: C

参考解析: 本题主要考查函数作为形参时的使用, 由 $x=8ss(funa, 9, 3)$; 可得到 $x=9+3=12$, 而后直接传入函数 $funb$, $x=x+8-3=12+5=17$ 。因此 C 选项正确。

18 有以下程序

```
#include <stdio.h>
main()
{
    char c[2][5] = {"6938", "8254"};
    *p[2];

    int i, j, s = 0;
    for(i = 0; i < 2; i++)
        for(j = 0; j < 5; j++)
            if(p[i][j] > 0; j += 2)
                s = 10 * s + p[i][j] - '0';
    printf("%d\n", s);
}
```

A. 9284B. 9824C. 6932D. 6385

参考答案: D

参考解析: 本题中首先是将二维字符数组 c 赋值指针数组 P , 然后遍历 P 找到两个字符串中的偶数位置的字符并组成一个数值输出, 根据代码执行即可得到结果为 6385。

19 设有定义语句:

```
char *aa[2] = {"abcd", "ABCD"};
```

则以下叙述正确的是

A. $aa[0]$ 存放了字符串 “abcd” 的首地址

B. aa 数组的两个元素只能存放含有 4 个字符的一维数组的首地址

C. aa 数组的值分别是字符串"abcd"和"ABCD"

D. fla 是指针变量，它指向含有两个元素的字符型数组

参考答案：A

参考解析：

本题中定义了一个指针数组，数组中存放的是字符指针，aa 数组的两个元素分别包含了 5 个字符，所以可以存放 5 个字符的一维数组的首地址，所以 B 选项错误。aa 数组的值是指向那两个字符串的字符指针，而不是那两个字符串，所以 C 选项错误。Eta 本身是一个指向两个字符指针的指针，所以 D 选项错误。

20 有如下定义

```
struct st
{
    int a; float b; char x[10];
} FILE *fp;
```

若文件已正确打开，且数组 x 的 10 个元素均已赋值，以下将数组元素写到文件中的语句错误的是

```
A) for(i=0; i<10; i++)
    fwrite(x, sizeof(struct st), 1, fp);
B) fwrite(x, 10, sizeof(struct st), 1, fp);
C) fwrite(x, sizeof(struct st), 10, fp);
D) for(i=0; i<10; i++)
    fwrite(&x[i], sizeof(struct st), 1, fp);
```

参考答案：A

参考解析：fwrite() 函数用法是 fwrite(void\$buffer, intsize, int n, FILE*fp)，其中 buffer 为要输出的数据的首地址，A 选项中，因为 n=1，即每次写入 1 个结构体数据，x 是数组的首地址，因此，每次写入的数据都是数组的首个结构体元素，没有将整个数组写入文件中去，答案为 A 选项。

21 若有以下程序

```

#include <stdio.h>
char f(char x)
{ return x * x + 'a'; }
main()
{ char a, b = 0;
  for ( a = 0; a < 4; a ++ )
    b = f(a); putchar(b); }

```

则程序的输出结果是

A. abedB. ABEJC. abejD. ABCD

参考答案: C

参考解析:

若有以下程序: 该题目首先初始化变量 a 和 b, 通过 for 循环语句使 a 取值为 0、1、2、3, f 函数的功能是将 a*a+'a' 的值返回给 b, 即 b 为 a...b e j。最后通过字符输出函数 putchar() 将 b 输出。因此 C 选项正确。

22 有以下程序

```

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

```

程序运行后的输出结果是

A. 17B. 19C. 18D. 20

参考答案: B

参考解析: 本题通过语句 P=&a[1] 将指针变量 P 指向了存储单元 a[1], 即使得 p[0] 的值为 4。然后通过 3 次 for 循环, x=0 时, *(P+x)=* P=a[1]=4 所以 y=1+4=5; x=1 时, *(P+x)=*(P+1)=a[2]=6, 所以 y=5+6=11; x=2 时, *(P+x)=*(P+2)=a[3]=8, 所以 y=11+8=19。因此 B 选项正确。

23 有以下程序

```

#include <stdio.h>
int *f(int *s,int *t)
{
    int *k;
    if (*s < *t) | k = s; s=t; t=k;

    return s;
}

main()
{
    int i=3, j=5, *p=&i, *q=&j, *r;
    r=f(p,q); printf("%d,%d,%d,%d,%d\n", i, j, *p, *q, *r);
}

```

则程序的输出结果是

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

参考答案: B

参考解析: 在主函数中分别给整型变量 i、j 和指针型变量 P、q 赋初值, 并声明指针变量 r; 调用 f 函数, 并将实参变量 P 和 q 的值传递给形参变量 s 和 t, 并且 f 函数是指针型函数, 即函数的返回值将是一个地址。在 f 函数中, 如果条件成立, 则将指针变量 s 和 t 互换, 并且将指针 s 的地址返回主函数。最后输出 i, j, *P, *q, *r 的值, 即 3, 5, 3, 5, 5。因此 B 选项正确。

24 以下针对全局变量的叙述错误的是

- A. 用 extern 说明符可以限制全局变量的作用域
- B. 全局变量是在函数外部任意位置上定义的变量
- C. 全局变量的作用域是从定义位置开始至源文件结束
- D. 全局变量的生存期贯穿于整个程序的运行期间

参考答案: A

参考解析：在不同编译单位内用 extern 说明符来扩展全局变量的作用域，extern 可以将全局变量作用域扩展到其他文件，而不是限制全局变量的作用域，答案为 A 选项。

25 以下叙述中正确的是

- A. 如果 P 是指针变量，则 &p 是不合法的表达式
- B. 如果 P 是指针变量，则 *P 表示变量 P 的地址值
- C. 在对指针进行加、减算术运算时，数字 1 表示 1 个存储单元的长度
- D. 如果 P 是指针变量，则 * P+1 和 *(P+1) 的效果是一样的

参考答案：C

参考解析：A 选项中，如果 P 是指针变量，则 &p 表示变量 P 的地址；B 选项中，如果 P 是指针变量，则 *P 表示变量 P 所指向的地址的值；D 选项中，如果 P 是指针变量，*P+1 表示将 P 所指的值加上 1，而 *(P+1) 表示的是先将指针右移一位再取所指向变量的值。因此 C 选项正确。

26 以下选项中错误的

- A. $a!=b$ 与 $a=a!b$ 等价
- B. $a\wedge b$ 与 $a\wedge b$ 等价
- C. $a|b$ 与 $a=a|b$ 等价
- D. $a\&b$ 与 $a=a\&b$ 等价

参考答案：A

参考解析：A 选项中， $a!=b$ 表示 a 不等于 b 时，运算结果为 1，或者为 0；而 $a=a!b$ 是一种语法错误， $!$ 是非运算，且是单目运算符，只要求有一个操作数，故两者不等价，答案为 A 选项。

27 有以下程序

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

程序的输出结果是

A. 1 5 0 B. 3 5 7 C. 1 4 7 D. 3 6 9

参考答案: B

参考解析:程序首先初始化二维数组X[3][3],然后通过3次for循环,输出x[0][2]、x[1][1]和x[2][0]的值,即3、5、7。因此B选项正确。

28 以下叙述中正确的是

- A. 只要适当地修改代码,就可以将do-while与while相互转换
- B. 对于“for(表达式1;表达式2;表达式3)循环体”首先要计算表达式2的值,以便决定是否开始循环
- C. 对于“for(表达式1;表达式2;表达式3)循环体”只在个别情况下才能转换成while语句
- D. 如果根据算法需要使用无限循环(即通常所称的“死循环”),则只能使用while语句

参考答案: A

参考解析: B选项中,for语句中需要先求解先求解表达式1;C选项中,for语句使用最为灵活,它完全可以取代while语句;D选项中,对于死循环,也可以使用for语句。因此A选项正确。

29 以下选项中函数形参不是指针的是

- A. fun(int*A, i {...})
- B. fun(int a[10]) {...}

C. fun(int&p) {...}

D. fun(int p[]) {...}

参考答案: C

参考解析: B 选项和 D 选项是将数组作为函数参数;A 选项是将指针作为函数参数;因此 C 选项正确。

30 若有定义和语句: int a, b;

scanf("%d, %d, ", &a, &b. ;

以下选项中的输入数据, 不能把值 3 赋给变量 a、5 赋给变量 b 的是

A. 3, 5, 4B. 3, 5C. 3, 5, D. 3, 5

参考答案: D

参考解析: scanf() 的格式控制串可以使用其他非空白字符, 如本题中的逗号, 但在输入时必须输入这些字符, 以保证匹配。所以在 3 和 5 之间必须输入逗号, 不能有其他符号, 才能保证 a 和 b 的正确赋值。所以选择 D。