

函数部分自测题

仅供班级内部学习，请勿外传

1. 在 C 语言程序中，有关函数的定义正确的是_____。

- A. 函数的定义可以嵌套，但函数的调用不可以嵌套
- B. 函数的定义不可以嵌套，但函数的调用可以嵌套
- C. 函数的定义和函数的调用均不可以嵌套
- D. 函数的定义和函数的调用均可以嵌套

解：函数的定义不能嵌套，也就是说不能在一个函数定义体内定义另外的函数，但函数调用可以嵌套，如递归函数。本题答案为 B。

2. 以下错误的描述是：函数调用可以_____。

- A. 出现在执行语句中
- B. 出现在一个表达式中
- C. 作为一个函数调用的实参
- D. 作为一个函数的形参

解：形参只能是变量，不能是函数调用，函数调用可以作为一个函数调用的形参。本题答案为 D。

3. 以下正确的函数定义形式是_____。

- A. void fun(int x,int y);
- B. void fun(int x;int y)
- C. void fun(int x,int y) {}
- D. void fun(int x,y);

解：void fun(int x,int y) {}是函数体为空的函数定义。注意本例指的是正确的函数定义而不是函数声明。本题答案为 C。

4. 有以下程序：

```
#include <stdio.h>
float fun(int x,int y)
{   return(x+y); }
void main()
{   int a=2,b=5,c=8;
    printf("%3.0f\n",fun((int)fun(a+c,b),a-c));
}
```

程序运行后的输出结果是_____。

- A. 编译出错
- B. 9
- C. 21
- D. 9.0

解：执行(int)fun(a+c,b)返回整数 15，执行 fun((int)fun(a+c,b),a-c)返回 9。本题答案为 B。

5. 有以下程序:

```
#include <stdio.h>
int f1(int x,int y)
{ return x>y?x:y; }
int f2(int x,int y)
{ return x>y?y:x; }
void main()
{ int a=4,b=3,c=5,d,e,f;
  d=f1(a,b);
  d=f1(d,c);
  e=f2(a,b);
  e=f2(e,c);
  f=a+b+c-d-e;
  printf("%d,%d,%d\n",d,f,e);
}
```

执行后输出结果是_____。

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

解: d=f1(a,b)返回 4, d=f1(d,c)返回 5, e=f2(a,b)返回 3, e=f2(e,c)返回 3, f=a+b+c-d-e=4。本题答案为 C。

6. 以下程序的输出结果是_____。

```
#include <stdio.h>
void fun(int x,int y)
{ x=x+y;y=x-y;x=x-y;
  printf("%d,%d",x,y);
}
void main()
{ int x=2,y=3;
  fun(x,y);
  printf("%d,%d\n",x,y);
}
```

解: 调用函数是直接参数传递。本题答案为 3,2,2,3。

7. 判断以下叙述的正确性。

- (1) 一个 C 程序可以由若干个函数组成, 这些函数可以书写在不同的文件中。
- (2) 函数定义可以嵌套, 函数调用也可以嵌套。
- (3) 所有的函数调用前都必须进行函数声明。
- (4) 函数声明中, 省略形参变量或者将形参变量写成其他名称, 不影响程序的正确性。
- (5) 编译器通过函数原型确定函数调用是否正确。
- (6) 一个 C 程序中只能包含一个 main()函数, 程序总是从 main()函数开始执行。
- (7) 函数的形参只能在函数内部使用。

- (8) C 语言中，实参可以是常量、变量或表达式。
- (9) C 语言中，实参的个数、类型和位置必须与形参的定义一一对应。
- (10) 如果函数值的类型和 `return` 语句中表达式的值不一致，则以函数类型为准。
- (11) 通过 `return` 语句，函数可以返回一个或一个以上的返回值。

解：(1) 正确。

- (2) 错误。函数定义不可以嵌套，函数调用可以嵌套。
- (3) 错误。如果一个函数定义出现在首次调用之前，则可以不声明。
- (4) 正确。
- (5) 正确。
- (6) 正确。
- (7) 正确。
- (8) 正确。
- (9) 正确。
- (10) 正确。
- (11) 错误。一个函数的 `return` 语句只能返回一个值。

8. 以下程序的输出结果是_____。

```
#include <stdio.h>
int fun(double a)
{   return a; }
void main()
{   double x=2.5;
    printf("j=%d\n",fun(x));
}
```

解：当从一个函数返回一个表达式的类型与函数返回类型不相同，返回的表达式类型自动转换成函数类型。本题答案为 `j=2`。

9. 以下叙述中错误的是_____。

- A. 在不同的函数中可以使用相同名称的变量
- B. 函数中的形式参数是局部变量
- C. 在一个函数内定义的变量只在本函数范围内有效
- D. 在一个函数内的复合语句中定义的变量在本函数范围内有效

解：在一个函数内的复合语句中定义的变量只在本复合语句的范围内有效。本题答案为 D。

10. 以下叙述中错误的是_____。

- A. 主函数中定义的变量在整个程序中都是有效的
- B. 在其他函数中定义的变量在主函数中也不能使用
- C. 形式参数也是局部变量
- D. 复合语句中定义的变量只在该复合语句中有效

解：主函数中定义的变量也是局部变量，只能在主函数中使用。本题答案为 A。

11. 将一个函数指定为 `static` 存储类别后，该函数将_____。

- A. 既可以被同一源文件中的函数调用，也可以被其他源文件中的函数调用
- B. 只能被同一源文件中的函数调用，不能被其他源文件中的函数调用

- C. 只能被其他源文件中的函数调用，不能被同一源文件中的函数调用
D. 既不能被同一源文件中的函数调用，也不能被其他源文件中的函数调用
解：静态函数只能被同一编译单元（源文件）中的函数所调用。本题答案为 B。

12. 设有以下函数：

```
#include<stdio.h>
f(int a)
{   int b=0;
    static int c=3;
    b++;c++;
    return(a+b+c);
}
```

如果在下面的程序中调用该函数，则输出结果是_____。

```
void main()
{   int a=2,i;
    for (i=0;i<3;i++)
        printf("%d",f(a));
}
```

- A. 7 8 9
B. 7 9 11
C. 7 10 13
D. 7 7 7

解：f 函数中的 c 为静态变量，第 1 次调用时 c 置初值 3，退出该函数时 c 不会释放存储空间，以后再调用 f 时，不再给 c 置初值，直接使用其以前的结果。本题答案为 A。

13. 以下程序的输出结果是_____。

```
#include<stdio.h>
int x=3;
void incre()
{
    static int x=1;
    x*=x+1;
    printf("%d ",x);
}
void main()
{   int i;
    for (i=1;i<x;i++)
        incre();
}
```

- A. 3 3
B. 2 2
C. 2 6
D. 2 5

解：函数外定义的 x 是全局变量，在 f 函数中定义的 x 为静态变量，第 1 次调用时 x 置初值

1, 退出该函数时 x 不会释放存储空间, 以后再调用 f 时, 不再给 x 置初值, 直接使用其以前的结果。全局变量 x 与 f 中的 x 没有冲突, 后者是静态局部变量。本题答案为 C。

14. 分析以下程序的执行结果。

```
#include <stdio.h>
int fun(int m)
{
    static int t=3;
    m+=t++;
    return(m);
}
void main()
{
    int m=2,i,j=0;
    for (i=0;i<2;i++)
        j+=fun(m++);
    printf("%d\n",j);
}
```

解: 在 `main` 函数中一共两次调用 `fun()` 函数。由于 `fun()` 函数中 t 是静态变量, 第一次调用 `fun(2)` 时将 t 置初值 3, 其形参 $m=2$, 执行 $m+=t++$; 后, $m=5$, $t=4$, 在退出函数执行时 t 不被释放, 返回 5。 $j=j+5=5$ 。第二次调用 `fun(3)` 时不再给 t 置初值, 保持上一次调用函数时的值, $m=3+t=3+4=7$, $t=5$, 返回 7。 $j=j+7=12$ 。程序输出为 12。

15. 以下程序输出结果有错, 指出错误的原因。

```
#include<stdio.h>
char *fun()
{
    char astr[]="ABCD";
    return astr;
}
void main()
{
    printf("%s\n",fun());
}
```

解: 在函数 `fun` 中, `astr` 是一个局部变量, 在 `fun` 函数执行完毕后, `astr` 就不存在了, 所以出错。只需将 `astr` 改为 `static` 变量即可。

16. 以下正确的说法是: 在 C 语言中_____。

- A. 实参与与其对应的形参各占用独立的存储单元
- B. 实参与与其对应的形参共占用一个存储单元
- C. 只有当实参与与其对应的形参同名时才共占用存储单元
- D. 形参是虚拟的, 不占用存储单元

解: A。

17. 以下程序的输出结果是_____。

```
#include <stdio.h>
int x=1;
```

```

void fun1()
{   int x;
    x=2;
    printf("%d ",x);
}
void fun2()
{   x+=2; }
void main()
{   printf("%d ",x);
    fun1();
    printf("%d ",x);
    fun2();
    printf("%d ",x);
    printf("\n");
}

```

- A. 1 2 1 3
- B. 1 1 2 2
- C. 1 2 2 3
- D. 1 2 3 3

解：函数外部定义的 **x** 是全局变量，**fun1** 中定义的 **x** 是局部变量（只在该函数中有效），在 **fun1** 中，由于同名，局部变量的 **x** 覆盖作为全局变量的 **x**。本题答案为 A。

18. 有如下程序：

```

#include <stdio.h>
void f(int x,int y)
{   int t;
    if(x<y)
    {   t=x;
        x=y;
        y=t;
    }
}
void main()
{   int a=4,b=3,c=5;
    f(a,b);
    f(a,c);
    f(b,c);
    printf("%d,%d,%d\n",a,b,c);
}

```

执行后输出结果是_____。

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

解：注意 f 函数的形参是普通整型变量，当 $x < y$ 时，x 和 y 的交换结果不会回传给实参。本题答案为 D。

19. 以下程序的输出结果是_____。

```
#include <stdio.h>
int n=10;
void fun()
{   int n=20;
    printf("%d",n);
}
void main()
{   fun();
    printf("%d\n",n);
}
```

- A. 10,20
- B. 10,10
- C. 20,10
- D. 20,20

解：这里 n 是全局变量，但在函数 fun 中定义了同名的局部变量，所以在函数中输出的是局部变量的值，在主函数中没有定义同名的变量，输出的是全局变量的值。本题答案为 C。

20. 有以下程序：

```
#include <stdio.h>
int a=2;
int f(int *a)
{   return (*a)++; }
void main()
{   int s=0;
    {   int a=5;
        s+=f(&a);
    }
    s+=f(&a);
    printf("%d\n",s);
}
```

执行后输出结果是_____。

- A. 10
- B. 9
- C. 7
- D. 8

解：这里的 a 为全局变量。本题答案为 C。

21. 以下 strcpy() 函数实现字符串的复制，即将 t 所指字符串复制到 s 所指向的内存空间中，形成一个新的字符串 s。请填空。

```

void sstrcpy(char *s,char *t)
{   while(*s++=____); }
void main()
{   char str1[100],str2[]="abcdefgh";
    sstrcpy(str1,str2);
    printf("%s\n",str1);
}

```

解：sstrcpy 函数的两个形参均为字符串指针。本题答案为*t++。

22. 以下程序的输出结果是_____。

```

#include <stdio.h>
void foo(int *a,int *b)
{   *a=*a+*b;
    *b=*a-*b;
    *a=*a-*b;
}
void main()
{   int a=1,b=2,c=3;
    foo(&a,&b);
    foo(&b,&c);
    foo(&c,&a);
    printf("%d,%d,%d\n",a,b,c);
}

```

解：函数 foo 的形参为传地址参数， $*a=*a+*b$ ， $*b=*a-*b=(*a+*b)-*b=*a$ ， $*a=*a-*b=(*a+*b)-*a=*b$ ，所以该函数的功能是将形参 a、b 的值交换。调用 foo(&a,&b)，交换 a、b 实参值，则 a=2，b=1，c=3；调用 foo(&b,&c)，交换 b、c 实参值，则 a=2，b=3，c=1；调用 foo(&c,&a)，交换 c、a 实参值，则 a=1，b=3，c=2。本题答案为 a=1,b=3,c=2。

23. 有以下程序：

```

#include <stdio.h>
int    *f(int *x,int *y)
{   if (*x<*y)
        return x;
    else
        return y;
}
void main()
{   int    a=7,b=8,*p;
    p=f(&a,&b);
    printf("%d\n",*p);
}

```

执行后输出结果是_____。

- A. 7
- B. 8

C. 0

D. 一个地址值

解：f 函数返回形参 x 和 y 所指整数中较小值的地址。本题答案为 A。

24. 以下程序的输出结果是_____。

```
#include <stdio.h>
char *f(char *b)
{   b+=3;
    return b;
}
void main()
{   char a[]="abcdef",*p;
    p=f(a);
    printf("%s\n",p);
}
```

解：函数 f 的形参数为字符串数组 a 的指针，执行函数时，将 b 后移 3 个元素，并返回该指针。本题答案为 def。

25. 下列程序的运行结果是_____。

```
#include <stdio.h>
#include <string.h>
char *ss(char *s)
{   return s+strlen(s)/2; }
void main()
{   char *p,*str="abcdefgh";
    p=ss(str);
    printf("%s\n",p);
}
```

解：ss 是指针型函数，返回形参 s 字符串的后半部分字符串。本题答案为 efgh。

26. 有以下程序：

```
#include <stdio.h>
int fa(int x)
{   return x*x; }
int fb(int x)
{   return x*x*x; }
int f(int (*f1)(),int (*f2)(),int x)    /*f1 和 f2 形参均为函数指针*/
{   return f2(x)-f1(x); }
void main()
{   int i;
    i=f(fa,fb,2);
    printf("%d\n",i);
}
```

程序运行后的输出结果是_____。

- A. -4
- B. 1
- C. 4
- D. 8

解: $i=f(fa,fb,2)=23-22=4$ 。本题答案为 C。

27. 有以下程序:

```
#include <stdio.h>
int f(int n)
{   if (n==1) return 1;
    else return f(n-1)+1;
}
void main()
{   int i,j=0;
    for(i=1;i<3;i++)
        j+=f(i);
    printf("%d\n",j);
}
```

程序运行后的输出结果是_____。

- A. 4
- B. 3
- C. 2
- D. 1

解: 这里的 f 是一个递归函数, $j=f(1)+f(2)=1+f(2)=1+f(1)+1=1+1+1+3$ 。本题答案为 B。

28. 请读程序:

```
#include<stdio.h>
void main(int argc,char *argv[])
{   int i;
    printf("%d ",argc);
    for (i=0;i<argc;i++)
        printf("%s ", argv[i]);
    printf("\n");
}
```

若上面的程序编译、连接后生成可执行文件 abc.exe, 则输入以下命令行

abc file1 file2

程序执行后的输出结果是_____。

解: 输出命令行参数个数和所有参数值。本题答案为 3 abc.exe file1 file2。