

一、函数

1. 单选

- 关于设计函数的目的，以下正确的说法是：_____。
 - A) 提高程序的执行效率
 - B) 提高程序的可读性
 - C) 减少程序的规模
 - D) 减少程序的内存占有量
- 在 C 语言程序中，有关函数的定义，下面_____是正确的。
 - 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都不可以嵌套
 - D) 函数的定义和函数的调用都可以嵌套
- 在 C 语言程序中，下面说法_____是正确的。
 - A) 对于用户自定义函数，在使用之前必须加以声明
 - B) 声明函数时不必指出其返回类型
 - C) 函数可以返回一个值，也可以什么值也不返回
 - D) 函数必须具备一定的功能，什么功能也没有的函数不合法
- 以下错误的说法是：_____。
 - A) 实参可以是常量、变量或表达式
 - B) 形参可以是常量、变量或表达式
 - C) 实参可以是任意类型
 - D) 形参应与其对应的实参类型一致
- 有关 C 语言程序的函数调用，以下错误的说法是：_____。
 - A) 函数调用可以出现在执行语句中
 - B) 函数调用可以出现在一个表达式中
 - C) 函数调用可以作为一个函数调用的实参
 - D) 函数调用可以作为一个函数调用的形参
- 以下正确的说法是：_____。
 - A) 函数的形参在函数未调用时预分配存储空间
 - B) 若函数的定义出现在主程序之前，则可以不必声明
 - C) `fun() {return 1;}` 函数定义是不允许的
 - D) 一般来说，函数的形参和实参的类型应该一致
- C 语言规定，函数返回值的类型是由_____决定。
 - A) `return` 语句中的表达式类型
 - B) 调用该函数时主调函数的类型
 - C) 调用该函数时的系统
 - D) 在定义该函数时所指定的函数类型
- 以下叙述中正确的是：_____。
 - A) 定义函数时，形参的类型定义可以放在函数体内
 - B) `return` 后面的值不能为表达式

- C) 如果函数类型与返回值类型不一致，以函数类型为准
D) 如果形参与实参类型不一致，以实参类型为准
- 以下正确的说法是：_____。
A) 调用函数时的实参必须是有确定值的变量
B) `return()`语句的括号中，可以是变量、常量或有确定值的表达式
C) C 语言中，函数调用时实参和形参间不能传递地址
D) 实参和形参的若类型不匹配，编译时报错
 - 若定义一个函数，且此函数中没有 `return` 语句，则正确的说法是：该函数_____。
A) 没有返回值
B) 返回若干个系统默认值
C) 能返回一个用户所希望的值
D) 返回一个不确定的值
 - 在 C 语言程序中，有关函数的声明，下面_____是正确的。
A) 对于自定义函数，在使用之前必须加以声明
B) 声明函数时不必指出其返回类型
C) 声明函数时，可以只指出形参类型，不必指出形参的名称
D) 对于库函数，不必声明即可直接使用
 - 函数调用语句 `func((exp1, exp2), (exp3, exp4, exp5));` 含有实参的个数为：_____。
A) 1
B) 2
C) 4
D) 5
 - 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)`
 - 以下函数值的类型是：_____。

```
fun(float x)
{
    float y;
    y=3*x-4;
    return y;
}
```


A) `int`
B) 不确定
C) `void`
D) `float`
 - 下面程序的输出结果是：_____。

```
#include <stdio.h>
int f(int a, int b)
{
    int c;
    if(a>b) c=1;
```

```

        else if(a==b) c=0;
        else c=-1;
        return c;
    }
int main()
{
    int i=2, p;
    p=f(i, i+1);
    printf("%d\n", p);
    return 0;
}

```

- A) -1
- B) 0
- C) 1
- D) 2

- 下面程序的输出结果是：_____。

```

#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;
}
int 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);
    return 0;
}

```

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

- 下面程序的输出结果是：_____。

```

#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;
    }
    int main()
    {
        int a=4, b=3, c=5, d, e, f;
        d=f1(f1(a, b), c);
        e=f2(f2(a, b), c);
        f=a+b+c-d-e;
        printf("%d, %d, %d\n", d, f, e);
        return 0;
    }

```

A) 3, 4, 5

B) 5, 3, 4

C) 5, 4, 3

D) 3, 5, 4

- 下面程序的输出结果是：_____。

```

#include <stdio.h>
float fun(int x, int y)
{
    return (x+y);
}
int main()
{
    int a=2, b=5, c=8;
    printf("%3.0f\n", fun((int)fun(a+c, b), a-c));
    return 0;
}

```

A) 编译出错

B) 9

C) 21

D) 9.0

- 下面程序的输出结果是：_____。

```

#include <stdio.h>
func(int a, int b)
{
    int c;
    c=a+b;
    return c;
}
int main()

```

```

{
    int x=6, y=7, z=8, r;
    r=func((x--, y++, x+y), z--);
    printf("%d\n", r);
    return 0;
}

```

- A) 11
- B) 20
- C) 21
- D) 31

- 以下对 C 语言函数的有关描述中，正确的是：_____。
 - A) 调用函数时，只能把实参的值传送给形参，形参的值不能传送给实参
 - B) C 函数既可以嵌套定义又可以递归调用
 - C) 函数必须有返回值，否则不能使用函数
 - D) C 程序中有调用关系的所有函数必须放在同一个源程序文件中
- 以下正确的说法是：在 C 语言中_____。
 - A) 实参和与其对应的形参各占用独立的存储单元
 - B) 实参和与其对应的形参共占用同一个存储单元
 - C) 只有当实参和与其对应的形参同名时才共用存储单元
 - D) 形参是虚拟的，不占用存储单元
- 以下程序的输出结果是：_____。

```

#include <stdio.h>
void f(int x, int y)
{
    int t;
    if(x<y)
    {
        t=x;
        x=y;
        y=t;
    }
}
int 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);
    return 0;
}

```

- A) 3,4,5
- B) 5,3,4
- C) 5,4,3

D) 4,3,5

- 以下程序的输出结果是：_____。

```
#include <stdio.h>
int cchar(char ch)
{
    if(ch<='Z' && ch>='A') ch -= 'A'-'a';
    return ch;
}
int main()
{
    char s[]="ABC+abc=defDEF", *p=s;
    while(*p)
    {
        *p=cchar(*p);
        p++;
    }
    printf("%s\n", s);
    return 0;
}
```

A) abc+ABC=DEFdef

B) abc+abc=defdef

C) abcaABCDEFdef

D) abcabdefdef

- 以下程序的输出结果是：_____。

```
#include <stdio.h>
void fun(char *c,int d)
{
    *c=*c+1;
    d=d+1;
    printf("%c,%c,",*c,d);
}
int main()
{
    char a='A',b='a';
    fun(&b,a);
    printf("%c,%c\n",a,b);
    return 0;
}
```

A) B,a,b,a

B) a,B,a,B

C) A,b,A,b

D) b,B,A,b

- 有以下语句，则错误的 fun 函数的首部是：_____。

```
int main()
```

```

{
    int a[50], n;
    .....
    fun(n, &a[9]);
    .....
    return 0;
}

```

- A) void fun(int m, int x[])
- B) void fun(int s, int h[41])
- C) void fun(int p, int *s)
- D) void fun(int n, int a)

- 程序运行后的输出结果是：_____。

```

#include <stdio.h>
#include <string.h>
char *f(char p[][10],int n)
{
    static char t[20];
    int i;
    unsigned max=0;
    for(i=0;i<n;i++)
        if(strlen(p[i])>max)
        {
            max(strlen(p[i]));
            strcpy(t,p[i]);
        }
    return t;
}
int main()
{
    char p[][10]={"abc","aabdfg","abbd","dcdbe","cd"};
    printf("%s\n",f(p,5));
    return 0;
}

```

- A) abc
- B) aabdfg
- C) cd
- D) dcdbe

- 若使用一维数组名作为实参，则以下叙述中正确的是：_____。

- A) 必须在主调函数中指定此数组的大小
- B) 实参数组类型与形参数组类型可以不匹配
- C) 在被调用函数中，不需要考虑形参数组的大小
- D) 实参数组名与形参数组名必须一致

- 若函数的形参为多维数组，则以下叙述正确的是：_____。

- A) 调用函数时实参数组的维数必须等于形参数组的维数

- B) 说明形参数组时可以省略每一维的大小
- C) 说明形参数组时只能省略第一维的大小
- D) 说明形参数组时必须指定每一维的大小
- 当调用函数时，实参是一个数组名，则向函数传送的是：_____。
 - A) 数组的长度
 - B) 数组的首地址
 - C) 数组每一个元素的地址
 - D) 数组每个元素中的值
- 以下程序的输出结果是：_____。

```
#include <stdio.h>
void sum(int *a)
{
    a[0]=a[1];
}
int main()
{
    int aa[10]={1,2,3,4,5,6,7,8,9,10}, i;
    for(i=2; i>=0; i--) sum(&aa[i]);
    printf("%d\n", aa[0]);
    return 0;
}
```

- A) 4
- B) 3
- C) 2
- D) 1
- 以下程序的输出结果是：_____。

```
#include <stdio.h>
void func(int *a, int b[])
{
    b[0] = *a+6;
}
int main()
{
    int a, b[5];
    a=0; b[0]=3;
    func(&a, b);
    printf("%d\n", b[0]);
    return 0;
}
```

- A) 6
- B) 7
- C) 8
- D) 9
- 以下程序运行时，从键盘上输入 `OPEN THE DOOR`，则它的输出结果是：_____。


```

#include <stdio.h>
char fun(char *c)
{
    if(*c<='Z' && *c>='A') *c -= 'A'-'a';
    return *c;
}
int main()
{
    char s[81], *p=s;
    gets(s);
    while(*p)
    {
        *p=fun(p);
        putchar(*p);
        p++;
    }
    putchar('\n');
    return 0;
}

```

A) oPEN tHE door

B) open the door

C) OPEN THE DOOR

D) Open The Door

- 以下程序的输出结果是：_____。

```

#include <stdio.h>
void fun(int *s)
{
    static int j=0;
    do
        s[j]+=s[j+1];
    while(++j<2);
}
int main()
{
    int k, a[10]={1,2,3,4,5};
    for(k=1; k<3; k++) fun(a);
    for(k=0; k<5; k++) printf("%d", a[k]);
    return 0;
}

```

A) 34756

B) 23445

C) 35745

D) 12345

- 以下程序的输出结果是：_____。

```

#include <stdio.h>
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;
            }
}
int main()
{
    int aa[10]={1,2,3,4,5,6,7,8,9,10}, i;
    sort(&aa[3], 5);
    for(i=0; i<10; i++) printf("%d,", aa[i]);
    printf("\n");
    return 0;
}

```

A) 1,2,3,4,5,6,7,8,9,10

B) 10,9,8,7,6,5,4,3,2,1

C) 1,2,3,8,7,6,5,4,9,10

D) 1,2,10,9,8,7,6,5,4,3

- 以下程序的输出结果是：_____。

```

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

```

A) 10

B) 18

C) 8

D) 15

- 以下程序的输出结果是：_____。

```
#include <stdio.h>
void ss(char *s, char t)
{
    while(*s)
    {
        if(*s==t) *s=t-'a'+'A';
        s++;
    }
}
int main()
{
    char str[100]="abcddfefdbd", c='d';
    ss(str, c);
    printf("%s\n", str);
    return 0;
}
```

- A) ABCDDFEFDBD
- B) abcDDfefDbD
- C) Abcddfefdbd
- D) abcddfefdbd

- 以下程序的输出结果是：_____。

```
#include <stdio.h>
int n;
void fun(int b[][3])
{
    n=b[1][1];
}
int main()
{
    int a[3][3]={1,2,3,4,5,6,7,8,9};
    fun(a);
    printf("%d\n",n);
    return 0;
}
```

- A) 1
- B) 4
- C) 7
- D) 5

- 以下程序中函数 f 的功能是将 n 个字符串按由大到小的顺序进行排序，则程序运行后的输出结果是：_____。

```
#include <stdio.h>
#include <string.h>
void f(char p[][10], int n)
```

```

{
    char t[20];
    int i, j;
    for(i=0; i<=n-1; i++)
        for(j=i+1; j<=n-1; j++)
            if(strcmp(p[i], p[j]) < 0)
            {
                strcpy(t, p[i]);
                strcpy(p[i], p[j]);
                strcpy(p[j], t);
            }
}

int main()
{
    char p[][10]={"abc", "aabdfg", "abbd", "dcdbe", "cd"};
    f(p, 5);
    printf("%d\n", strlen(p[0]));
    return 0;
}

```

- A) 6
- B) 4
- C) 5
- D) 3

- 程序中有如下的数组定义和 f()函数调用语句：

```

int a[3][4];
f(a);

```

则在 f()函数的定义中，对形参数组 array 的正确定义方式为：_____。

- A) f(int array[][6])
- B) f(int array[3][])
- C) f(int array[][4])
- D) f(int array[2][5])

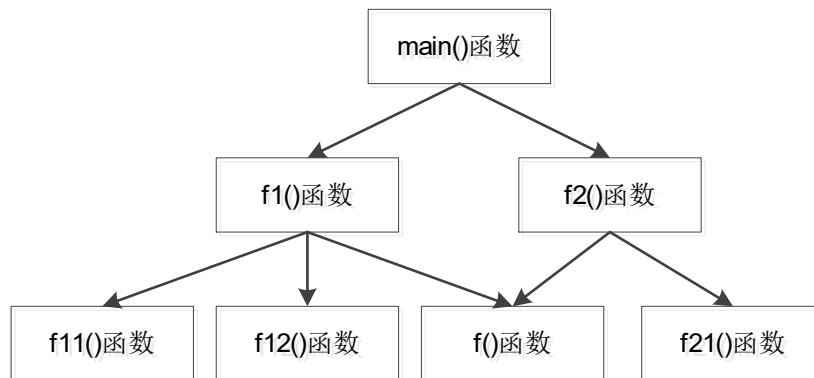
2. 判断

- 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。
- 函数定义可以嵌套，函数调用也可以嵌套。
- 所有的函数调用前都必须进行函数声明。
- 函数声明中，省略形参变量名或者将形参变量写成其他名称，不影响程序的正确性。
- 函数的形参只能在函数内部使用。
- C 语言中，实参可以是常量、变量或表达式。
- 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。
- 编译器通过函数原型确定函数调用是否正确。
- 通过 return 语句，函数可以返回一个或一个以上的返回值。
- 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。

- 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。

3. 简答

- 如图是一个 C 程序中的 `main` 函数及所有自定义函数之间的调用关系（箭头指向被调用函数，箭尾指向主调函数），程序执行过程中，图中各函数的执行顺序是自上而下、自左到右的。如果在程序的一次执行过程中，图中所有被调用函数都被它的主调函数调用一次并且只调用一次。试给出这次执行过程中，图中各函数“拥有”程序控制权的先后次序。



4. 程序运行结果

- 下面程序的输出结果是：_____。

```

#include <stdio.h>
void fun(int x, int y)
{
    x=x+y;
    y=x-y;
    x=x-y;
    printf( "%d,%d,",x,y);
}
int main()
{
    int x=2, y=3;
    fun(x, y);
    printf( "%d,%d\n",x,y);
    return 0;
}
  
```

- 下面程序的输出结果是：_____。

```

#include <stdio.h>
int fun(int a, int b)
{
    printf("%d,%d\n",a,b);
    return(a>b?a:b);
}
  
```

```

}
int main()
{
    int i=2,j=5,k=3,m;
    m=fun(fun(i,j),fun(j,k));
    printf("%d\n",m);
    return 0;
}

```

- 下面程序的输出结果是：_____。

```

#include <stdio.h>
int fun(double a)
{
    return a;
}
int main()
{
    double x=2.5;
    printf("j=%d\n", fun(x));
    return 0;
}

```

- 分析以下程序的执行结果：

```

#include <stdio.h>
void func(char str[])
{
    int i, a, b;
    for(a=b=0; str[a] != '\0'; a++)
        if(str[a] != 'c') str[b++] = str[a];
    str[b] = '\0';
}
int main()
{
    char str[] = "abcdef";
    func(str);
    printf("str[] = %s\n", str);
    return 0;
}

```

5. 程序完形填空

- 以下 fun 函数求 x 的 y 次方，请在空白处填上恰当的程序元素。

```

double fun(double x, int y)
{
    int i;
    double z = 1.0;

```

```

        for(i=0; i (1) ; i++) z= (2) ;
    return z;
}

```

- 下列函数用于确定一个给定字符串 str 的长度，请填空。

```

int strlen(char str[])
{
    int num=0;
    while( (1) ) ++num;
    return( (2) );
}

```

- 下列函数 inverse 的功能是使一个字符串按逆序存放，请填空。

```

void inverse(char str[])
{
    char m;
    int i, j;
    for(i=0, j=strlen(str); i< (1) ; i++, (2) )
    {
        m = str[i];
        str[i] = (3) ;
        (4) ;
    }
}

```

- 下面 fun 函数的功能是将形参 x 的值转换成二进制数，所得二进制数的每一位数放在一维数组中返回，二进制数的最低位放在下标为 0 的元素中，其他依此类推。请填空。

```

void fun(int x, int b[])
{
    int k=0, r;
    do
    {
        r = x% (1) ;
        b[k++] = r;
        x /= (2) ;
    } while(x);
}

```

- 下面 search()函数的功能是：采用二分查找法，在已按字母次序从小到大排序的字符数组 list[0..len-1]中，查找字符 c，若 c 在数组中，函数返回字符 c 在数组中的下标，否则返回-1。

```

int search(char list[], char c, int len)
{
    int low=0, high=len-1, k;
    while( (1) )
    {
        k=(low+high)/2;

```

```

        if( (2) ) return k;
        else if( (3) ) high=k-1;
        else low=k+1;
    }
    return -1;
}

```

- 下面 search()函数的功能是：采用二分查找法，在已“升序”的字符数组 list 中，查找字符 c，若 c 在数组中，函数返回字符 c 在数组中的下标，否则返回-1。注意：函数

int lenstr(char s[]);返回的函数值是字符串 s 的长度。

```

int search(char str[],char c)
{
    int lenstr(char s[]);
    int low=0,high,k;
    high=lenstr(str)-1;
    while( (1) )
    {
        k=(low+high)/2;
        if(str[k]==c) return (2);
        else if(str[k]>c) high=k-1;
        else (3)=k+1;
    }
    return -1;
}

```

二、变量的作用域与存储类别

1. 单选

- 以下说法中错误的是：_____。
 - 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
- 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能在该复合语句中有效
- 有关 C 语言的全局变量，以下正确的说法是：_____。
 - A) 全局变量在定义它的文件中的任何地方都是有效的
 - B) 全局变量在程序的全部执行过程中一直占用内存单元
 - C) 全局变量不能与局部变量共用同一个标识符
 - D) 全局变量有利于程序的模块化和可读性的提高

- 在一个 C 源文件中定义的全局变量的作用域是：_____。
 - A) 所在源文件的全部范围
 - B) 所在 C 程序的全部范围
 - C) 所在 C 函数的全部范围
 - D) 所在源文件中，自定义位置起到文件结束止
- 以下叙述中正确的是：_____。
 - A) 全局变量的作用域一定比局部变量的作用域范围大
 - B) 静态存储方式的变量其生存期贯穿于整个程序的运行期间
 - C) 函数的形参都属于全局变量
 - D) 未在定义语句中赋初值的 `auto` 变量和 `static` 变量的初值都是随机值
- 以下程序的输出结果是：_____。

```
#include <stdio.h>
f(int a)
{
    int b=0;
    static int c=3;
    b++;
    c++;
    return a+b+c;
}
int main()
{
    int a=2,i;
    for(i=0;i<3;i++) printf("%d ",f(a));
    printf("\n");
    return 0;
}
```

- A) 7 8 9
 - B) 7 9 11
 - C) 7 10 13
 - D) 7 7 7
- 以下程序的输出结果是：_____。

```
#include <stdio.h>
n=5;
int f()
{
    static int n=0;
    int s=1;
    s+=n;
    n++;
    return s;
}
int main()
{
```

```

int i, a=0;
for(i=0; i<n; i++) a+=f();
printf("%d\n", a);
return 0;
}

```

- A) 20
- B) 24
- C) 25
- D) 15

- 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);
}
int main()
{
    int i;
    for(i=1; i<x; i++) incre();
    return 0;
}

```

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

- 以下程序的输出结果是：_____。

```

#include <stdio.h>
f(int a)
{
    int b=0;
    static int c=3;
    a=c++; b++;
    return (a);
}
int main()
{
    int a=2, i, k;
    for(i=0; i<2; i++) k=f(a++);
    printf(“%d\n”, k);
    return 0;
}

```

A) 3

B) 0

C) 5

D) 4

- 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;
}
int main()
{
    printf("%d ", x);
    fun1();
    printf("%d ", x);
    fun2();
    printf("%d ", x);
    printf("\n");
    return 0;
}
```

A) 1 2 1 3

B) 1 1 2 2

C) 1 2 2 3

D) 1 2 3 3

- 以下程序的输出结果是：_____。

```
#include <stdio.h>
int a, b;
void fun()
{
    a=100;
    b=200;
}
int main()
{
    int a=5, b=7;
    fun();
    printf("%d%d\n", a, b);
}
```

```
    return 0;
}
```

- A) 100200
- B) 57
- C) 200100
- D) 75

- 以下程序的输出结果是：_____。

```
#include <stdio.h>
int n=10;
void fun()
{
    int n=20;
    printf("%d", n);
}
int main()
{
    fun();
    printf("%d\n", n);
    return 0;
}
```

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

- 以下程序的输出结果是：_____。

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

- A) 4
- B) 6
- C) 8
- D) 10

- 以下程序的输出结果是：_____。

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

- A) 4
- B) 6
- C) 8
- D) 10

- 以下程序的输出结果是：_____。

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

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

2. 判断

- C 语言中变量只有 `extern` 和 `static` 两种存储类别说明符。

- 变量的存储类别确定了变量在内存中存在的时间。
- C 语言中的函数只有 `extern` 和 `static` 两种存储类别说明符。
- 变量根据其作用域的范围可以分为局部变量和全局变量。
- 全局变量在整个程序执行期间保留其值。
- 一个函数中的 `static` 变量在退出该函数时仍保持其值。
- `static` 局部变量只能在本函数中使用。
- C 语言要求一个程序的所有编译单元中的函数名不能相重。
- 在一个编译单元中调用另一编译单元中的函数与调用本编译单元中的函数没有任何差别。
- 一个函数的定义必须完整地存在于同一个文件中，而不能把它分散在两个或多个文件中。
- 在 C 语言中使用系统函数库中的函数时要用 `#include` 包含该函数的头文件。

3. 简答

- 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;
}
int main()
{
    printf("%s\n", fun());
    return 0;
}
```

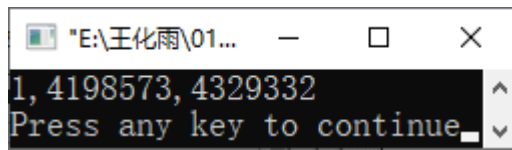
- 源程序：

```
#include <stdio.h>
int *getArray()
{
    int arr[3]={1,2,3};
    return arr;
}
int main()
{
    int i, *arrPtr=getArray();
    for(i=0;i<3;i++)
        printf("%d%c",arrPtr[i],i!=2?' ':'\n');
    return 0;
}
```

- 上面源程序编译时出现警告信息：warning C4172: returning address of local variable or temporary

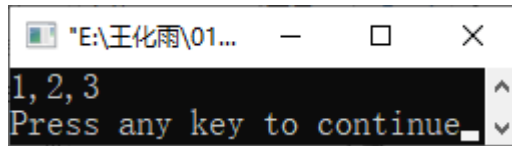
✧ 定位该警告信息，光标指向 `getArray` 函数的“返回语句” `return arr;`

- ✧ 程序的运行结果是：



```
"E:\王化雨\01...  -  □  ×  
1, 4198573, 4329332  
Press any key to continue_
```

- ✧ 而不是理想的：



```
"E:\王化雨\01...  -  □  ×  
1, 2, 3  
Press any key to continue_
```

- 请解释上述情况的原因；并给出源程序的“改正”方法。

4. 程序运行结果

- 以下程序的输出结果是：_____。

```
#include <stdio.h>  
int func(int a, int b)  
{  
    static int m=0, i=2;  
    i+=m+1;  
    m=i+a+b;  
    return m;  
}  
int main()  
{  
    int k=4, m=1, p;  
    p=func(k, m); printf("%d, ", p);  
    p=func(k, m); printf("%d\n", p);  
    return 0;  
}
```

- 以下程序的输出结果是：_____。

```
#include <stdio.h>  
int fac(int n)  
{  
    static int f=1, t=0, s=0;  
    int i;  
    if(n>t)  
        for(i=t+1; i<=n; i++) f=f*i;  
    else  
    {  
        f=1;  
        for(i=1; i<=n; i++) f=f*i;  
    }  
    t=n;  
}
```

```

        s=s+f;
        return(s);
    }
int main()
{
    int a[]={3, 5, 2}, i, sum;
    for(i=0; i<3; i++) sum=fac(a[i]);
    printf("sum=%d\n", sum);
    return 0;
}

```

- 分析以下程序的执行结果。

```

#include <stdio.h>
void fun()
{
    int x=1;
    static int y=2;
    x++; y++;
    printf("%d, %d\n", x, y);
}
int main()
{
    fun();
    fun();
    printf("\n");
    return 0;
}

```

- 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);
}
int main()
{
    int m=2, i, j=0;
    for(i=0; i<2; i++) j+=fun(m++);
    printf("%d\n", j);
    return 0;
}

```