

## 一、函数的参数传递

### 1. 单选

#### 1.1 函数参数传递

- 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) abcabcdefdef

- 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

## 1.2 数组名作为函数参数

- 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

- 实参向形参进行数据传递时，数据传递的方向是单向的，即形参变量值的改变不影响实参变量的值。
- 函数形参的存储单元是动态分配的。
- 数组名作为函数的参数时，实参数组和形参数组共用相同的内存单元。

## 3. 简答

- 某班有 3 位学生（序号分别是 0、1、2），各学 3 门课，下面的程序用来输出第 n 个学生的成绩，请改正其中的错误。



```
#include <stdio.h>
void search(float *p[3], int n)
{
    int i;
    printf("the scores of NO.%d are: ", n);
    for(i=0; i<3; i++)
        printf("%g ", *(p+n-1)+i);
    printf("\n");
}
int main()
{
    float score[3][3]={65,67,79},{80,86,84},{50,90,76}};
    search(*score, 2);
    return 0;
}
```

- C 程序中，定义了下面 3 个函数，请问这 3 个函数在参数传递方面各有什么差别？简述理由。

- (1) `int f1(int p){ return p++; }`
- (2) `int f2(int *p){ return *(p++); }`
- (3) `int f3(int *p){ return (*p)++; }`

- 正常情况下，以下程序的输出结果是 5。

- 但由于有一个带标号的语句有错，程序的运行结果不正确。
- 请指出有错误语句的标号，并说明错误的原因。

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

#### 4. 程序运行结果

- 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;
}
int 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);
    return 0;
}

```

- 以下程序的输出结果是：\_\_\_\_\_。

```

#include <stdio.h>
void testf(int *p)
{
    *p=0;
}
int main()
{
    int *n,m[2];
    n=m;
    m[0]=1;
    m[1]=8;
    testf(n);
    printf("%d\n", *n);
    return 0;
}

```

- 以下程序的输出结果是：\_\_\_\_\_。

```

#include <stdio.h>
void testf(int **p)
{
    (*p)++;
}
int main()
{
    int *n,m[2];
    n=m;
    m[0]=1;
    m[1]=8;
    testf(&n);
    printf("%d\n",*n);
    return 0;
}

```

```
}
```

## 5. 程序完形填空

- 以下函数用来求出两整数之和，并通过形参将结果传回，请填空。

```
void func(int x, int y, _____ z)
{
    *z=x+y;
}
```

- 设在主函数中有以下定义和函数调用语句，且 fun 函数为 void 类型。请写出 fun 函数的首部\_\_\_\_\_。要求形参名为 b。

```
int main()
{
    double s[10][22];
    int n;
    ...
    fun(s);
    ...
    return 0;
}
```

- 以下程序中，select 函数的功能是：在 N 行 M 列的二维数组中，选出一个最大值作为函数值返回，并通过形参传回此最大值所在的行下标。请填空。

```
#include <stdio.h>
#define N 3
#define M 3
int select(int a[N][M], int *n)
{
    int i, j, row=1, col=1;
    for(i=0; i<N; i++)
        for(j=0; j<M; j++)
            if(a[i][j]>a[row][col])
                {
                    row=i;
                    col=j;
                }
    *n = (1) ;
    return (2) ;
}

int main()
{
    int a[N][M]={9, 11, 23, 6, 1, 15, 9, 17, 2}, max, n;
    max=select(a, &n);
    printf("max=%d, line=%d\n", max, n);
    return 0;
}
```

```
}
```

- 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 strcpy(char *s, char *t)
```

```
{
```

```
    while(*s++ = _____);
```

```
}
```

```
int main()
```

```
{
```

```
    char str1[100], str2[]="abcdefgh";
```

```
    strcpy(str1, str2);
```

```
    printf("%s\n", str1);
```

```
    return 0;
```

```
}
```

## 二、指针型函数

### 1. 单选

- 以下程序的输出结果是：\_\_\_\_\_。

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

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

```
{
```

```
    if(*x < *y) return x;
```

```
    else return y;
```

```
}
```

```
int main()
```

```
{
```

```
    int a=7, b=8, *p;
```

```
    p=f(&a, &b);
```

```
    printf("%d\n", *p);
```

```
    return 0;
```

```
}
```

A) 7

B) 8

C) 0

D) 一个地址值

### 2. 程序运行结果

- 以下程序的输出结果是：\_\_\_\_\_。

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

```
char *f(char *b)
```

```

{
    b+=3;
    return b;
}
int main()
{
    char a[]="abcdef", *p;
    p=f(a);
    printf("%s\n", p);
    return 0;
}

```

- 以下程序的输出结果是：\_\_\_\_\_。

```

#include <stdio.h>
#include <string.h>
char *ss(char *s)
{
    return s+strlen(s)/2;
}
int main()
{
    char *p, *str="abcdefgh";
    p=ss(str);
    printf("%s\n", p);
    return 0;
}

```

### 三、指向函数的指针

#### 1. 单选

- 设有定义 `int (*ptr)();`，则以下叙述中正确的是：\_\_\_\_\_。
  - A) ptr 是指向一维数组的指针变量
  - B) ptr 是指向 int 型数据的指针变量
  - C) ptr 是指向函数的指针，该函数是无参函数并返回一个 int 型数据
  - D) ptr 是一个无参函数的函数名，该函数的返回值是指向 int 型数据的指针
- 设有定义 `int *f();`，则标识符 f 代表的是：\_\_\_\_\_。
  - A) 一个用于指向整型数据的指针变量
  - B) 一个用于指向一维数组的行指针
  - C) 一个用于指向函数的指针变量
  - D) 一个返回值为指针型的函数名
- 以下程序的输出结果是：\_\_\_\_\_。
 

```

#include <stdio.h>
int fa(int x)

```

```

    {
        return x*x;
    }
int fb(int x)
{
    return x*x*x;
}
int f(int (*f1)(int), int (*f2)(int), int x)
{
    return f2(x)-f1(x);
}
int main()
{
    int i;
    i=f(fa, fb, 2);
    printf("%d\n", i);
    return 0;
}

```

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

## 2. 简答

- 理解以下 C 定义的含义：
  - (1) void (\*a)(int);
  - (2) void \*b(int);
  - (3) float (\*c(void))[6];
  - (4) double (\*d[6])(void);
- 以下声明都代表什么？
  - (1) float (\*\*def)[10];
  - (2) double \*(gh)[10];
  - (3) double (\*f[10])();
  - (4) int \*((\*b)[10]);
  - (5) long (\*fun)(int);
  - (6) int ((\*f)(int, int))(int);

## 3. 程序运行结果

- 以下程序的输出结果是：\_\_\_\_\_。

```

#include <stdio.h>
int funa(int a,int b)
{

```

```

    return a+b;
}
int funb(int a,int b)
{
    return a-b;
}
int sub(int (*t)(int,int),int x,int y)
{
    return (*t)(x,y);
}
int main()
{
    int x,(*p)(int,int);
    p=funa;
    x=sub(p,9,3);
    p=funb;
    x+=sub(p,8,3);
    printf("%d\n",x);
    return 0;
}

```

- 在执行以下程序时输入 10 和 20，分析其输出结果。

```

#include <stdio.h>
void input(int *p,int *q)
{
    printf("输入： ");
    scanf("%d%d",p,q);
}
void add(int *p,int *q)
{
    printf("输出： %d\n",*p+*q);
}
int main()
{
    int x,y;
    void (*b)(int*,int*);
    b=input;
    (*b)(&x, &y);
    b=add;
    (*b)(&x, &y);
    return 0;
}

```

#### 4. 程序完形填空

- 设函数 findbig 已定义为求 3 个数中的最大值，以下程序将利用函数指针调用 findbig 函数，请填空。

```
#include <stdio.h>
int main()
{
    int findbig(int,int,int);
    int (*f)(int,int,int),x,y,z,big;
    f = _____;
    scanf("%d%d%d",&x,&y,&z);
    big=(*f)(x,y,z);
    printf("big=%d\n",big);
    return 0;
}
```