

# C语言综合研究与高强度程序设计训练10

阅读下面的内容，理解每一个细节，然后构造程序来描述函数指针数组的用法和向函数传函数指针的方法

关于函数指针

指针的值是地址，函数指针的值是函数的入口地址

下面与 `int` 型变量对比，说明函数指针变量的定义方法

`int a;` 整型变量a

`int*a;` 整型指针变量a

`int a(char, char);` 函数 a，函数类型为:返回值为 `int`，参数为 `char`、`char`

`int (*a)(char, char);` 返回值为 `int`，参数为 `char``char` 的函数的函数指针变量a

可见类型加 `*` 则表示指向某种类型的指针。

为什么 `int*a` 直接加 `*`，而 `int (*a)(char, char)` 要用括号？

如果不加括号：`int*a(char, char)` 是返回 `int*` 的函数。为了不和函数 `int*a(char, char)` 混淆加上括号，`int (*a)(char, char)` 表示一个指向函数的指针。

下面与 `int` 型对比，说明函数指针类型的描述方法：

`int a;` 整型变量a

`int` 整型

`int (*a)(char, char);` 函数指针a

`int (*)(char, char);` 函数指针 a 返回值为 `int` 型，参数为 `char`、`char` 的函数的函数指针

函数指针使用举例：

## • 程序1

```
1  int f(char, char);
2  main() {
3      int a;
4      int b;
5      long c;
6      int (*p)(char, char);
7
8      p = f;
9      b = (int)f;
10     c = (long)f;
11
12     printf("main=%x f=%x\n", main, f);
13     printf("p=%x    b=%x c=%lx\n", p, b, c);
14
15     a = p(1, 2);
16     printf("%d\n", a);
```

```

17
18     a = ((int (*)(char, char))b)(1, 2);
19     printf("%d\n", a);
20 }
21 int f(char a, char b) { return a + b; }

```

- 程序2

```

1  int far f(char, char);
2  main() {
3      int a;
4      int b;
5      long c;
6      int(far * p)(char, char);
7
8      p = f;
9      b = (int)f;
10     c = (long)f;
11
12     printf("main offset=%x\n", main);
13     printf("main=%lx\n", (long)main);
14     printf("f offset=%x\n", (int)f);
15     printf("f=%lx\n", f);
16     printf("p=%lx b=%x c=%lx\n", p, b, c);
17
18     a = p(1, 2);
19     printf("%d\n", a);
20
21     a = ((int(far * p)(char, char))c)(1, 2);
22     printf("%d\n", a);
23 }
24 int far f(char a, char b) { return a + b; }

```

## 向函数传函数指针

- code

```

1  int add(int a, int b) { return a + b; }
2  int f2(int (*p)(int, int));
3  int a;
4  main() {
5      p = add;
6      a = f2(p);
7      printf("%d\n", a);
8  }
9  int f2(int (*p)(int, int)) {
10     printf("-----f2-----\n");
11     return p(1, 2) + 1;
12 }

```

- 运行结果

```
C:\>\SRC\010_TEN\C.EXE
```

```
-----f2-----
```

```
4
```

```
C:\>_
```

## 函数指针数组的用法

- code

```
1  int add(int a, int b) { return a + b; }
2  int sub(int a, int b) { return a - b; }
3  int mul(int a, int b) { return a * b; }
4  int div(int a, int b) { return a / b; }
5  int f2(int (*p)(int, int));
6  void f3(int (*p[])(int, int), int);
7  int (*p)(int, int);
8  int (*pb[4])(int, int) = {add, sub, mul, div};
9  int a;
10 int i;
11 main() {
12     p = add;
13     a = f2(p);
14     printf("%d\n", a);
15     f3(pb, 4);
16 }
17 int f2(int (*p)(int, int)) {
18     printf("-----f2-----\n");
19     return p(1, 2) + 1;
20 }
21 void f3(int (*p[])(int, int), int size) {
22     printf("-----f3-----\n");
23     for (i = 0; i < size; i++) {
24         a = p[i](8, 4);
25         printf("%d\n", a);
26     }
27 }
```

- 运行结果

```
C:\>\SRC\010_TEN\C.EXE
```

```
-----f2-----
```

```
4
```

```
-----f3-----
```

```
12
```

```
4
```

```
32
```

```
2
```

```
C:\>_
```