

函数与指针

指针函数

含义 一个函数可以返回各种类型的数据，也可以返回一个指针类型的数据，返回指针值的函数被称为指针函数

定义 函数类型 *函数名(参数1, 参数2, ..., 形参n)

注意 指针函数中不可以返回局部变量的地址，因为局部变量地址的内存函数调用结束后会被系统回收，从而而出现段错误

```
1 #include <stdio.h>
2
3 int *point_fun()
4 {
5     static int a = 100;
6
7     return &a;
8 }
9
10 int main(int argc, const char *argv[])
11 {
12     int *a = point_fun();
13
14     printf("a = %d\n", *a);
15
16     return 0;
17 }
```

总结 总体原则是：返回的指针所对应的内存空间不能因该指针函数的返回而被释放掉

- 1、函数中动态分配的内存的首地址
- 2、函数中静态变量或全局变量所对应的存储单元的首地址
- 3、通过指针形参所获得的实参的有效地址

练习 使用指针函数实现将一串字符串中的小写字符转换为大写，指针函数实现转换，主函数实现打印

```
1 #include <stdio.h>
2
3 char *change_capital(char *p)
4 {
5     char *q = p;
6
7     while(*q != '\0')
8     {
9         if (*q >= 'a' && *q <= 'z')
10             *q -= 32;
11         q++;
12     }
13
14     return p;
15 }
16
17 int main(int argc, const char *argv[])
18 {
19     char a[] = "123abc";
20
21     char *p = change_capital(a);
22
23     printf("%s\n", p);
24
25     return 0;
26 }
```

函数指针

含义 函数指针是专门用来存放函数地址的指针，函数地址是一个函数的入口地址，函数名代表了函数的入口地址

定义

形式 数据类型 (*函数指针变量名)(参数列表)

举例

int (*p)(int a, int b);

数据类型 函数所指向的函数的返回值类型
函数指针变量名 满足用户标识符命名规则
参数列表 应与指向的函数的形参列表一致

2. 定义一个函数指针求出两个数中的最大值

```
1 #include <stdio.h>
2
3 int find_max(int a, int b)
4 {
5     return a > b ? a : b;
6 }
7
8 int main(int argc, const char *argv[])
9 {
10     int (*p)(int a, int b);
11
12     p = find_max;
13
14     printf("max = %d\n", p(3, 1));
15
16     return 0;
17 }
```

练习

1. 定义一个函数指针实现两个数的加、减、乘、除

```
1 #include <stdio.h>
2
3 int num_add(int a, int b)
4 {
5     return a + b;
6 }
7
8 int num_sub(int a, int b)
9 {
10     return a - b;
11 }
12
13 int num_mul(int a, int b)
14 {
15     return a * b;
16 }
17
18 int num_div(int a, int b)
19 {
20     return a / b;
21 }
22
23 int main(int argc, const char *argv[])
24 {
25     int (*p)(int a, int b);
26     int a = 0, b = 0;
27
28     printf("please input two data:\n");
29     scanf("%d%d", &a, &b);
30
31     p = num_add;
32     printf("add = %d\n", p(a, b));
33
34     p = num_sub;
35     printf("sub = %d\n", p(a, b));
36
37     p = num_mul;
38     printf("mul = %d\n", p(a, b));
39
40     p = num_div;
41     printf("div = %d\n", p(a, b));
42
43     return 0;
44 }
```