

C语言：计算器

将每一个运算都封装为一个函数，再用指针调用。

此计算器可以一次性的计算 加 减 乘 除 按位左移 按位右移 按位与 按位或 按位异或 等计算

运用了 do while语句 switch语句 分支语句 函数调用 函数指针 递归 的一些知识：

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

void menu()
{
    printf("*****\n");
    printf("***    1->'+'    2->'-'    ***\n");
    printf("***    3->'*'    4->'/'    ***\n");
    printf("***    5->'>>'    6->'<<'    ***\n");
    printf("***    7->'&'    8->'|'    ***\n");
    printf("***          9->'^'          ***\n");
    printf("*****\n");
    printf("***          0.exit          ***\n");
    printf("*****\n");
}

int add(int x, int y)
{
    return x + y;
}

int sub(int x, int y)
{
    return x - y;
}

int mul(int x, int y)
{
    return x * y;
}

int divs(int x, int y)
{
    return x / y;
}

int Shift_right_bit(int x, int y)
{
    return x >>= y;
}

int Shift_left_bit(int x, int y)
{

```

```
    return x <= y;
}

int Bitwise_with(int x, int y)
{
    return x & y;
}

int Bitwise_or(int x, int y)
{
    return x | y;
}

int Bitwise_XOR(int x, int y)
{
    return x ^ y;
}

void Print_Binary(unsigned int x)
{
    if (x > 1)
    {
        Print_Binary(x >> 1);
    }
    putchar((x & 1) ? '1' : '0');
}

// void calu(int (*pf)(int, int))
//{
//    int x = 0;
//    int y = 0;
//    int z = 0;
//    printf("请输入操作数");
//    scanf("%d %d", &x, &y);
//    z = pf(x, y);
//    printf("十进制结果:%d\n", z);
//    printf("二进制结果:\n");
//    Print_Binary(z);
//    printf("\n");
//}

void calus(int (*pfs)(int,int))
{
    int x = 0;
    int y = 0;
    int z = 0;
    printf("请输入操作数");
    scanf("%d %d", &x, &y);
    z = pfs(x,y);
    // 以二进制输出
    //      plana: 用库函数itoa
    //      planb: 手“冻”实现
    printf("十进制结果:%d\n",z);
    printf("二进制结果:");
```

```
    Print_Binary(z);
    printf("\n");
}

int main()
{
    menu();
    int input = 0;
    do
    {
        printf("请输入: ");
        scanf("%d",&input);
        switch (input)
        {
            case 1:
                calus(add);
                break;
            case 2:
                calus(sub);
                break;
            case 3:
                calus(mul);
                break;
            case 4:
                calus(divs);
                break;
            case 5:
                calus(Shift_right_bit);
                break;
            case 6:
                calus(Shift_left_bit);
                break;
            case 7:
                calus(Bitwise_with);
                break;
            case 8:
                calus(Bitwise_or);
                break;
            case 9:
                calus(Bitwise_XOR);
                break;
            case 0:
                printf("《退出》");
                break;
            default:
                printf("输入错误! 重新输入! ");
                break;
        }
    } while (input);

    return 0;
}
```

如果有什么错误，欢迎指出，如果有帮助，点个赞，谢谢。