arm的复习题

直接看asm，那个c语言看不懂

```
if( (key1()+key2()+key3()) == key ){
        printf("Congratz!\n");
        int fd = open("flag", O_RDONLY);
        char buf[100];
        int r = read(fd, buf, 100);
        write(0, buf, r);
}
else{
```

可以看到三个函数的和相加 = key

分别看3个函数

```
Dump of assembler code for function key1:
   0x00008cd4 <+0>:     push    {r11}           ; (str r11, [sp, #-4]!)
   0x00008cd8 <+4>:     add     r11, sp, #0
   0x00008cdc <+8>:     mov     r3, pc
   0x00008ce0 <+12>:    mov     r0, r3
   0x00008ce4 <+16>:    sub     sp, r11, #0
   0x00008ce8 <+20>:    pop     {r11}           ; (ldr r11, [sp], #4)
   0x00008cec <+24>:    bx      lr
```

返回值是r0，即pc（程序计数器），注意在arm32中，pc总是指向当前正在运行的指令后的第二条

即key1 = 0x8cdc+8

```
   0x00008cf0 <+0>:     push    {r11}           ; (str r11, [sp, #-4]!)
   0x00008cf4 <+4>:     add     r11, sp, #0
   0x00008cf8 <+8>:     push    {r6}            ; (str r6, [sp, #-4]!)
   0x00008cfc <+12>:    add     r6, pc, #1
   0x00008d00 <+16>:    bx      r6
   0x00008d04 <+20>:    mov     r3, pc
   0x00008d06 <+22>:    adds    r3, #4
   0x00008d08 <+24>:    push    {r3}
   0x00008d0a <+26>:    pop     {pc}
   0x00008d0c <+28>:    pop     {r6}            ; (ldr r6, [sp], #4)
   0x00008d10 <+32>:    mov     r0, r3
   0x00008d14 <+36>:    sub     sp, r11, #0
   0x00008d18 <+40>:    pop     {r11}           ; (ldr r11, [sp], #4)
   0x00008d1c <+44>:    bx      lr
```

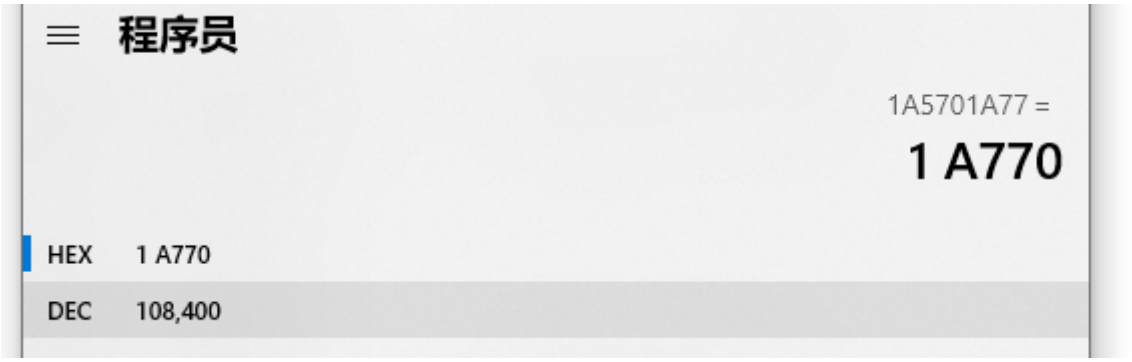
第二个函数有点特殊，r6 = pc+1 = 0x8cfc+8+1，注意，因为arm32指令pc的最低位bit是用来表示转换状态的，当执行bx r6时，程序跳转到0x8d04，同时最低为清0。此时，mov r3，pc ，r3 = 0x8d04+4

add r3，#4后，r3 = 0x8d04+8

```
(gdb) disass key3
Dump of assembler code for function key3:
   0x00008d20 <+0>:     push    {r11}           ; (str r11, [sp, #-4]!)
   0x00008d24 <+4>:     add     r11, sp, #0
   0x00008d28 <+8>:     mov     r3, lr
   0x00008d2c <+12>:    mov     r0, r3
   0x00008d30 <+16>:    sub     sp, r11, #0
   0x00008d34 <+20>:    pop     {r11}           ; (ldr r11, [sp], #4)
   0x00008d38 <+24>:    bx      lr
```

lr是保留返回地址的，可以知道是0x8d80

得到



108400是答案