2 实验内容

- do_loop 背景提要

```
12 int main()
13 {
14     short x, y, k;
15     if (scanf("%hd%hd%hd", &x, &y, &k))
16         printf("%hd\n", do_loop(x, y, k));
17     return 0;
18 }
```

2.1.2 实验内容

1. 输入 x = 2, y = 4000, k = 3，使用 gdb 进入 do_loop 的第一次循环，观察寄存器的值
   1) 在执行指令 cltd 前 %edx 的值是多少？
      0x3　3
   2) 在刚执行完 cltd 后 %edx 的值是多少？
      0x0　0
   3) 在执行指令 idiv 后 %edx 的值又变为了多少？请解释这种变化。
      0x1　1，idiv 将%rdx 和%rax 中的 128 位数作为被除数，即 4000，
      除数为 3，商储存在%rax 中，为 1333，余数储存在%rdx 中，为 1。
2. 使用输入 x = 2, y = 40000, k = 3 重复 (a) 的内容
   1）0x3　3
   2）0xffffffff　4294967295
   3）0x0　0，short 表示的范围是-32768~32767，40000 超出该范围
      %rax 寄存器的值符号扩展到%rdx，得到被除数为 0xfffffffffff9c40 即-25536
      除数为 3，商储存在%rax 中，为-8512，余数储存在%rdx 中，为 0。
3. 请回答 cltd 指令的作用
   将%rax 寄存器的值符号扩展到%rdx 寄存器，共同构成 idvi 指令的被除数。

- if-else 背景提要

2.1.3 实验内容

修改 if_else.s 中 if_else 片段，只允许修改分支条件，达到如下要求。
(A) 输入 12 15 ，要求现在 if_else 的返回值为 1（原来返回值为 0）
(B) 输入学号后四位，（如学号后四位是 1234 则输入 12 34）要求输出结果为 2

```
river@ubuntu:~/Desktop/workspace/lab03$ ./if_else_A
12 15
1
river@ubuntu:~/Desktop/workspace/lab03$ ./if_else_B
01 54
2
```

- Switch 背景提要

2.1.4 实验内容

1. 分别输入参数：
   - n =3;　　4
   - n =6;　　15
   - n =9;　　12
   - n =12;　　53
   - n =13;　　13
   - n =14;　　28

   利用 gdb，观察每个输入后 switchCase 函数的返回值各是多少？

2. 根据可执行文件 switch 的汇编代码，将如下 switchCase 的代码填写完整：

```
int switchCase(int n) {
    int result = 0;
    switch (n) {
        case 3:
            result = (n >> 1);
            break;
        case 6:
            result = n + n – 3;
            break;
        case 8:
            result = (n >> 2) + 1;
            break;
        case 9:
            result = (n >> 2) + 1;
            break;
        case 10:
            result = n + n + n + 5;
            break;
        case 12:
            result = n + n + n + 5;
            break;
        case 13:
            result = ((n + (n >> 31)) >> 1) – 6;
            break;
        default:
            result = n;
    }
    result += n;
    return result;
}
```