

# 实验报告——lab5 Riddle

PB20151793 宋玮

## 1.实验过程：

```
int judge(int r0) {  
    int i = 2;  
    r1 = 1;  
    while (i * i <= r0) {  
        if (r0 % i == 0) {  
            r1 = 0;  
            break;  
        }  
        i++;  
    }  
    return r1;  
}
```

首先，完成初始化内容，即 R1=1; i=2;

**ADD R1,R1, #1**

**ADD R2,R2, #2**

接着，跳转到子程序（JUDGE）执行，执行完成后返回并结束程序（halt）

**JSR JUDGE**

**HALT**

### 子程序部分：

- (1) 对辅助寄存器 R4, R5 进行初始化

**JUDGE ADD R4,R2, #-1**

**ADD R5,R2, #0**

其中，JUDEG 为子程序标号

- (2) 计算  $i*i$

**AGAIN ADD R5,R5, R2**

**ADD R4,R4, #-1**

**BRp AGAIN**

- (3) 比较  $i*i$  与 R0 大小关系

先对  $i*i$ （结果存储在 R5 中）取反加一，得到  $-i*i$

再加上 R0，即表示  $R0 - i*i$

结果若为负数，说明  $R0 < i*i$ ，则直接返回主程序

其他情况，则继续执行下面的指令。

**NOT R5,R5**

**ADD R5,R5, #1**

**ADD R5,R0,R5**

**BRn RETURN**

(4) 将 R0 值赋给辅助寄存器 R3, 并对 R2 求反加一, 存入 R6

```
ADD R3,R0, #0
ADD R6,R2, #0
NOT R6,R6
ADD R6,R6, #1
```

(5) 判断 R0 中的值是否能整除 i

若能整除, 则跳转至 TRUE 部分, 令 R1 等于 0, 并返回主程序  
若不能整除, 则对 R2 加一, 并重复执行子程序 (JUDGE)

```
AGAIN2 ADD R3,R3,R6
      BRz TRUE
      BRp AGAIN2
      ADD R2,R2, #1
      BRnzp JUDGE
```

(6) RETURN RET

(7) TRUE 部分

```
TRUE AND R1,R1, #0
      RET
```

## 2.总结:

完整程序如下:

```
.ORIG x3000
ADD R1,R1, #1
ADD R2,R2, #2
JSR JUDGE
HALT
JUDGE ADD R4,R2, #-1
      ADD R5,R2, #0
AGAIN ADD R5,R5, R2
      ADD R4,R4, #-1
      BRp AGAIN
      NOT R5,R5
      ADD R5,R5, #1
      ADD R5,R0,R5
      BRn RETURN
      ADD R3,R0, #0
      ADD R6,R2, #0
      NOT R6,R6
      ADD R6,R6, #1
AGAIN2 ADD R3,R3,R6
      BRz TRUE
      BRp AGAIN2
      ADD R2,R2, #1
      BRnzp JUDGE
RETURN RET
TRUE AND R1,R1, #0
      RET
.END
```

在 LC-3 评测上证实了该程序正确性。