

Lab 4 实验报告

- 姓名：谭旭
- 学号：PB20030775

实验内容

编写一 LC3 程序，实现判断素数。

实验步骤

为判断一个数是否为素数，需要使用不大于其平方根的所有数除该数，判断余数是否为 0，

若存在一个数使得计算结果为 0，则该数不是素数，

反之，若所有数都不能使得余数为 0，则该数是素数。

如下程序

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

考虑到平方根的计算较为困难，乘法也需要使用大量加法完成，因此选择降低程序的性能，判断所有小于原数的数能否整除原数（不包括 1）。

其 c++ 代码为

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

```

    }
    return r1;
}

```

取模部分

取模运算需要先实现除法。由于本次实验中不需要记录商，可以考虑使用除数循环减去被除数，当被除数被减为非正数的时候：

- 如果结果为负数，则被除数不能被除数整除
- 如果结果为 0，则被除数被除数整除

代码

```

MINUS    ADD R3, R3, R2
          BRp MINUS
          BRz NOREMAIN
          BRn REMAIN

```

循环部分

循环部分需要取所有 $i, 2 \leq i < n$ 进行取模运算。

考虑到如果 n 为偶数且大于 2，一定存在 2 可以整除 n ，

如果 n 为奇数，所有偶数都不能整除 n ，

所以只需要计算 $n - 2, n - 4, n - 6, \dots$ 是否能整除 n 即可。

代码

```

          NOT R2, R0
          ADD R2, R2, #1
LOOP      ADD R2, R2, #2
          ADD R4, R2, #2
          BRp PRIME
          ADD R3, R0, #0
MINUS     ADD R3, R3, R2
          BRp MINUS
          BRz NPRIME
          BRn LOOP

```

上述代码中，首先对 R0 取负数，存放到 R2。

每次循环都要判断 R2 是否大于 -2 ，若大于，则证明循环结束，R0 是素数；否则 R2 继续取下一个值，继续进行取模运算。

判断素数

完整代码如下

```
        .ORIG    x3000
        JSR JUDGE
        HALT
JUDGE   AND R1, R1, #0
        ADD R1, R1, #1
        NOT R2, R0
        ADD R2, R2, #1
LOOP    ADD R2, R2, #2
        ADD R4, R2, #2
        BRp PRIME
        ADD R3, R0, #0
MINUS   ADD R3, R3, R2
        BRp MINUS
        BRz NPRIME
        BRn LOOP
NPRIME  AND R1, R1, #0
PRIME   RET
        .END
```

将多余语句删去后，代码如下

```
        .ORIG    x3000
        JSR JUDGE
        HALT
JUDGE   AND R1, R1, #0
        ADD R1, R1, #1
        NOT R2, R0
        ADD R2, R2, #1
LOOP    ADD R2, R2, #2
        ADD R4, R2, #2
        BRp FINAL
        ADD R3, R0, #0
MINUS   ADD R3, R3, R2
        BRp MINUS
        BRn LOOP
        AND R1, R1, #0
FINAL   RET
        .END
```