中国科学技术大学计算机学院
《计算机系统概论实验报告》

实验题目：Riddle

学生姓名：叶子昂

学生学号：PB20020586

完成时间：2021年12月26日

0.1 实验目的：

> 将**C**程序翻译为**LC3**程序

0.2 实验环境：

- window11 和 wsl2
- VScode以及LC3TooL

0.3 实验过程：

0.3.1 几个核心问题

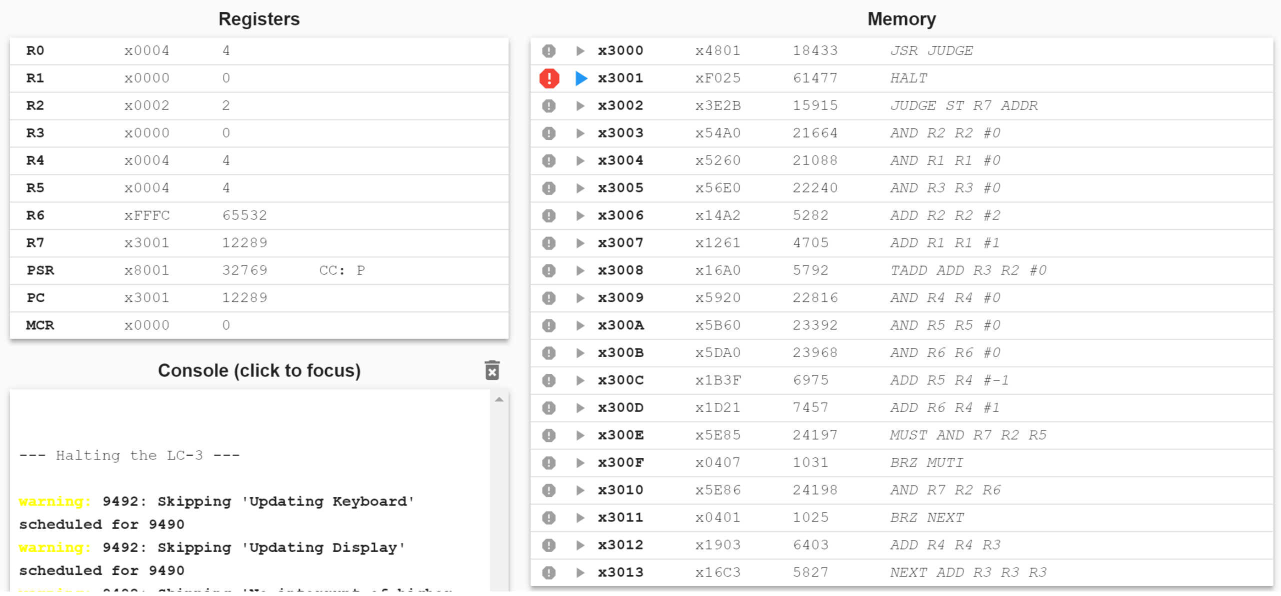
- 实现$i \times i$, 可以通过使用lab1中的代码，为减少指令数使用p版本的乘法代码。
- 判断r_0对i的模是否为0，判断模是否为0相对于求出余数来说简单不少，对2，4，8...的余数求起来较为简单，但对其他数求余数则较为复杂，本次实验仅需判断余数是否为0，故采用累加判断法——反复累加i直至与r_0相等或大于r_0。
- 实验中将判断余数为0单独作为一个子函数，存在多次调用需要注意保存r_7。

0.3.2 代码实现

```
.ORIG X3000
JSR JUDGE
HALT

JUDGE    ST R7 ADDR
           AND R2 R2 #0
           AND R1 R1 #0
           AND R3 R3 #0
           ADD R2 R2 #2;i=2
           ADD R1 R1 #1;R1=1
          ;**************************
TADD      ADD R3 R2 #0;while(i*i<=r0)
           AND R4 R4 #0;求i的平方
           AND R5 R5 #0
           AND R6 R6 #0
           ADD R5 R4 #-1
           ADD R6 R4 #1
MUST      AND R7 R2 R5
           BRZ MUTI
           AND R7 R2 R6
           BRZ NEXT
           ADD R4 R4 R3
NEXT      ADD R3 R3 R3
           ADD R5 R5 R5
           ADD R6 R6 R6
           BRNZP MUST
          ;*****************
MUTI      NOT R3 R0     ;求-R0
           ADD R3 R3 #1
           ADD R5 R4 R3X
           BRP OUT0;
          ;****************************
           JSR MOD0;    if(r0%i==0)
          ;****************************
           BRZ RE1
           ADD R2 R2 #1; i++
           BRNZP TADD; BR back to get i*i
          ;****************************
RE1       AND R1 R1 #0;set zero
OUT0      LD R7 ADDR
           RET
         ;*****************
MOD0      AND R5 R5 #0  ;判断r0%i是否为0
           AND R6 R6 #0
           NOT R6 R0
           ADD R6 R6 #1
BACK      ADD R5 R5 R2;反复累加i
           BRNZ RES;越界判断
           ADD R3 R5 R6
           BRZP OUT1;set condition作为返回结果
           BRNZP BACK
RES       AND R3 R3 #0
           ADD R3 R3 #1
OUT1      RET
ADDR     .FILL #0
.END
```

0.3.3 测试结果

输入4：



Registers

R0	x0004	4	
R1	x0000	0	
R2	x0002	2	
R3	x0000	0	
R4	x0004	4	
R5	x0004	4	
R6	xFFFC	65532	
R7	x3001	12289	
PSR	x8001	32769	CC: P
PC	x3001	12289	
MCR	x0000	0	

Console (click to focus)

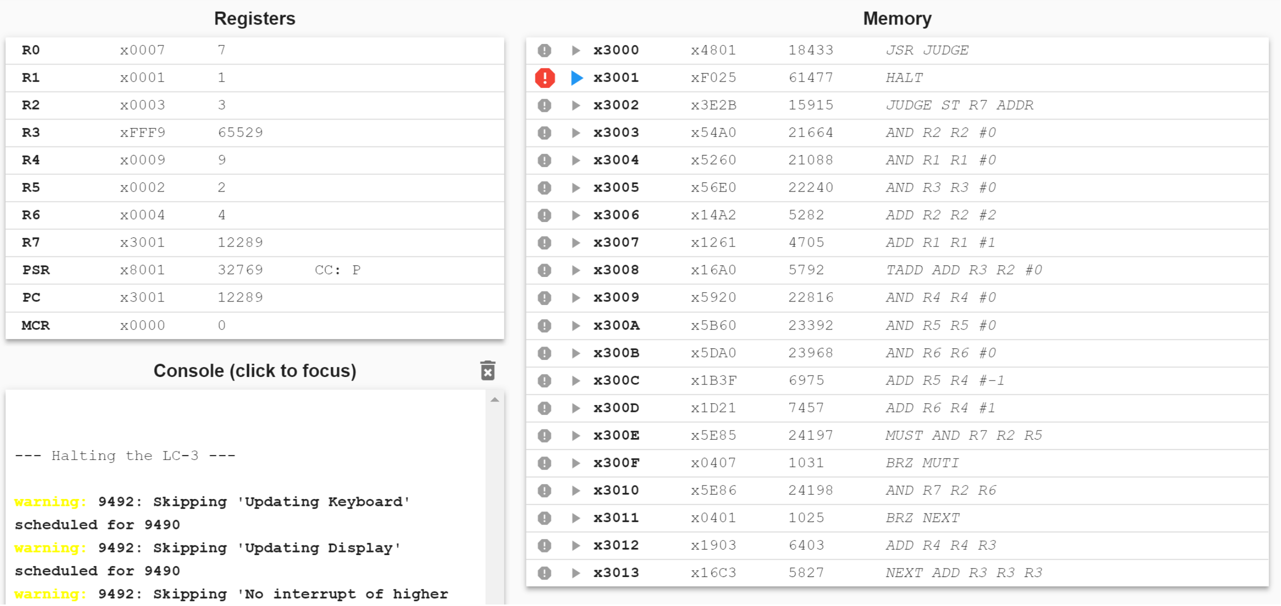
```
--- Halting the LC-3 ---

warning: 9492: Skipping 'Updating Keyboard'
scheduled for 9490
warning: 9492: Skipping 'Updating Display'
scheduled for 9490
```

Memory

x3000	x4801	18433	JSR JUDGE
x3001	xF025	61477	HALT
x3002	x3E2B	15915	JUDGE ST R7 ADDR
x3003	x54A0	21664	AND R2 R2 #0
x3004	x5260	21088	AND R1 R1 #0
x3005	x56E0	22240	AND R3 R3 #0
x3006	x14A2	5282	ADD R2 R2 #2
x3007	x1261	4705	ADD R1 R1 #1
x3008	x16A0	5792	TADD ADD R3 R2 #0
x3009	x5920	22816	AND R4 R4 #0
x300A	x5B60	23392	AND R5 R5 #0
x300B	x5DA0	23968	AND R6 R6 #0
x300C	x1B3F	6975	ADD R5 R4 #-1
x300D	x1D21	7457	ADD R6 R4 #1
x300E	x5E85	24197	MUST AND R7 R2 R5
x300F	x0407	1031	BRZ MUTI
x3010	x5E86	24198	AND R7 R2 R6
x3011	x0401	1025	BRZ NEXT
x3012	x1903	6403	ADD R4 R4 R3
x3013	x16C3	5827	NEXT ADD R3 R3 R3

输出为0。

输入7：



Registers

R0	x0007	7	
R1	x0001	1	
R2	x0003	3	
R3	xFFF9	65529	
R4	x0009	9	
R5	x0002	2	
R6	x0004	4	
R7	x3001	12289	
PSR	x8001	32769	CC: P
PC	x3001	12289	
MCR	x0000	0	

Memory

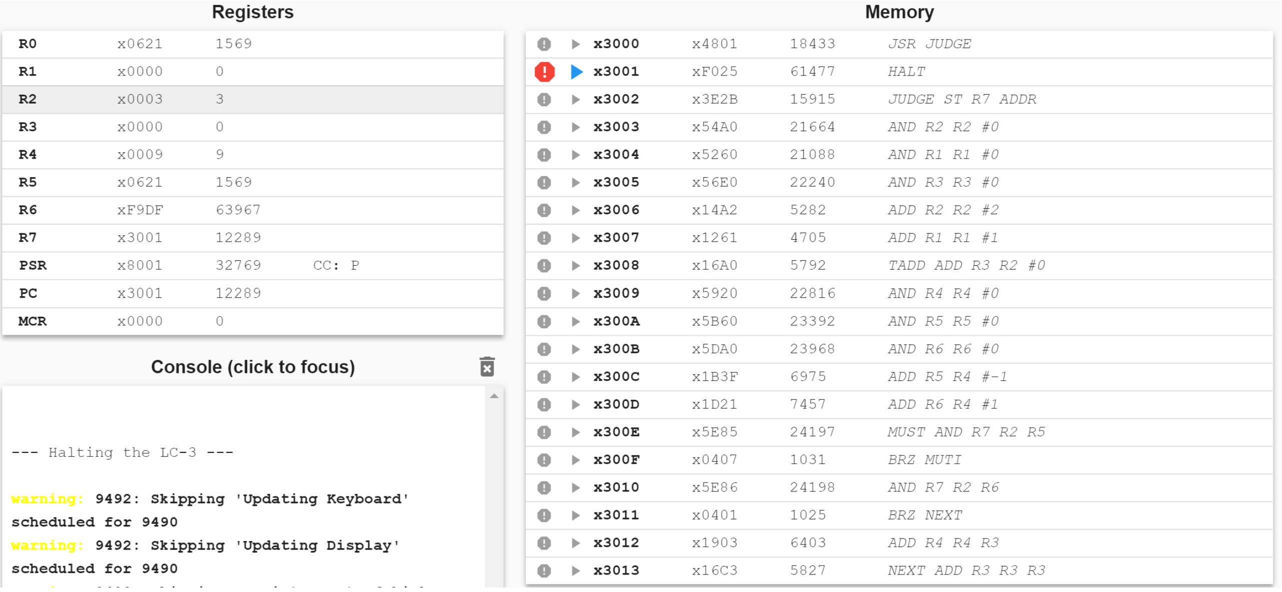
x3000	x4801	18433	JSR JUDGE
x3001	xF025	61477	HALT
x3002	x3E2B	15915	JUDGE ST R7 ADDR
x3003	x54A0	21664	AND R2 R2 #0
x3004	x5260	21088	AND R1 R1 #0
x3005	x56E0	22240	AND R3 R3 #0
x3006	x14A2	5282	ADD R2 R2 #2
x3007	x1261	4705	ADD R1 R1 #1
x3008	x16A0	5792	TADD ADD R3 R2 #0
x3009	x5920	22816	AND R4 R4 #0
x300A	x5B60	23392	AND R5 R5 #0
x300B	x5DA0	23968	AND R6 R6 #0
x300C	x1B3F	6975	ADD R5 R4 #-1
x300D	x1D21	7457	ADD R6 R4 #1
x300E	x5E85	24197	MUST AND R7 R2 R5
x300F	x0407	1031	BRZ MUTI
x3010	x5E86	24198	AND R7 R2 R6
x3011	x0401	1025	BRZ NEXT
x3012	x1903	6403	ADD R4 R4 R3
x3013	x16C3	5827	NEXT ADD R3 R3 R3

Console (click to focus)

```
--- Halting the LC-3 ---

warning: 9492: Skipping 'Updating Keyboard'
scheduled for 9490
warning: 9492: Skipping 'Updating Display'
scheduled for 9490
warning: 9492: Skipping 'No interrupt of higher
```

输出为1。

输入1569：



Registers

R0	x0621	1569	
R1	x0000	0	
R2	x0003	3	
R3	x0000	0	
R4	x0009	9	
R5	x0621	1569	
R6	xF9DF	63967	
R7	x3001	12289	
PSR	x8001	32769	CC: P
PC	x3001	12289	
MCR	x0000	0	

Memory

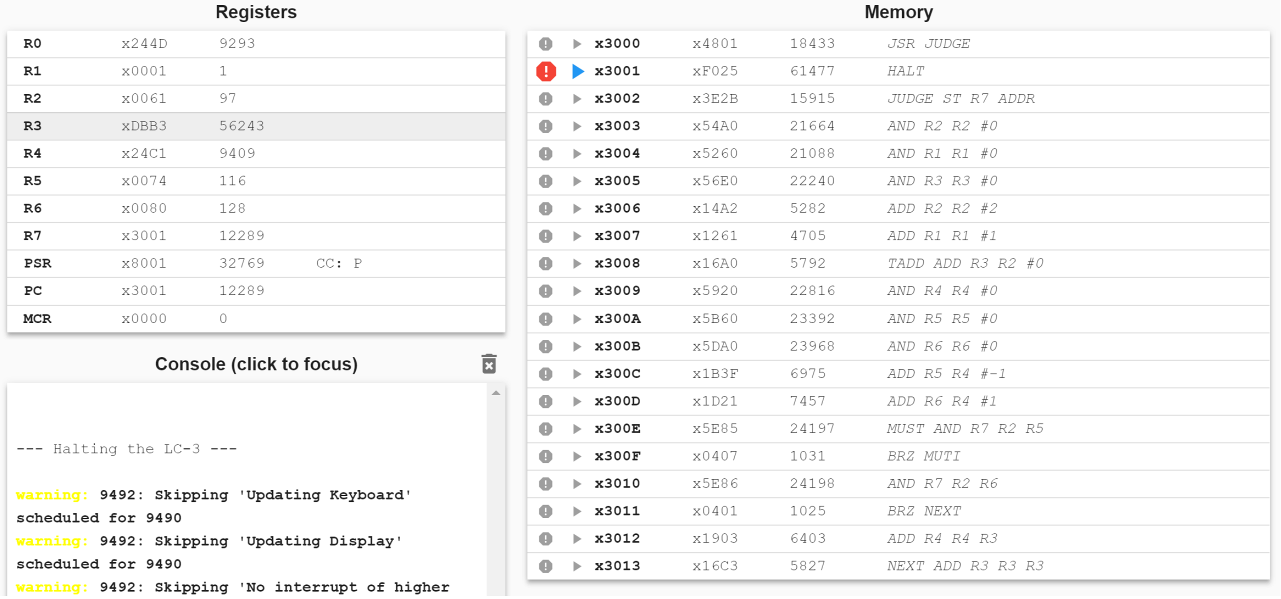
x3000	x4801	18433	JSR JUDGE
x3001	xF025	61477	HALT
x3002	x3E2B	15915	JUDGE ST R7 ADDR
x3003	x54A0	21664	AND R2 R2 #0
x3004	x5260	21088	AND R1 R1 #0
x3005	x56E0	22240	AND R3 R3 #0
x3006	x14A2	5282	ADD R2 R2 #2
x3007	x1261	4705	ADD R1 R1 #1
x3008	x16A0	5792	TADD ADD R3 R2 #0
x3009	x5920	22816	AND R4 R4 #0
x300A	x5B60	23392	AND R5 R5 #0
x300B	x5DA0	23968	AND R6 R6 #0
x300C	x1B3F	6975	ADD R5 R4 #-1
x300D	x1D21	7457	ADD R6 R4 #1
x300E	x5E85	24197	MUST AND R7 R2 R5
x300F	x0407	1031	BRZ MUTI
x3010	x5E86	24198	AND R7 R2 R6
x3011	x0401	1025	BRZ NEXT
x3012	x1903	6403	ADD R4 R4 R3
x3013	x16C3	5827	NEXT ADD R3 R3 R3

Console (click to focus)

```
--- Halting the LC-3 ---

warning: 9492: Skipping 'Updating Keyboard'
scheduled for 9490
warning: 9492: Skipping 'Updating Display'
scheduled for 9490
```

输出为0。

输入9293：



Registers

R0	x244D	9293	
R1	x0001	1	
R2	x0061	97	
R3	xDBB3	56243	
R4	x24C1	9409	
R5	x0074	116	
R6	x0080	128	
R7	x3001	12289	
PSR	x8001	32769	CC: P
PC	x3001	12289	
MCR	x0000	0	

Memory

x3000	x4801	18433	JSR JUDGE
x3001	xF025	61477	HALT
x3002	x3E2B	15915	JUDGE ST R7 ADDR
x3003	x54A0	21664	AND R2 R2 #0
x3004	x5260	21088	AND R1 R1 #0
x3005	x56E0	22240	AND R3 R3 #0
x3006	x14A2	5282	ADD R2 R2 #2
x3007	x1261	4705	ADD R1 R1 #1
x3008	x16A0	5792	TADD ADD R3 R2 #0
x3009	x5920	22816	AND R4 R4 #0
x300A	x5B60	23392	AND R5 R5 #0
x300B	x5DA0	23968	AND R6 R6 #0
x300C	x1B3F	6975	ADD R5 R4 #-1
x300D	x1D21	7457	ADD R6 R4 #1
x300E	x5E85	24197	MUST AND R7 R2 R5
x300F	x0407	1031	BRZ MUTI
x3010	x5E86	24198	AND R7 R2 R6
x3011	x0401	1025	BRZ NEXT
x3012	x1903	6403	ADD R4 R4 R3
x3013	x16C3	5827	NEXT ADD R3 R3 R3

Console (click to focus)

```
--- Halting the LC-3 ---

warning: 9492: Skipping 'Updating Keyboard'
scheduled for 9490
warning: 9492: Skipping 'Updating Display'
scheduled for 9490
warning: 9492: Skipping 'No interrupt of higher
```

输出为1。
均正确。

程序行数：49行
指令执行数：在labS中测试上述数据结果为38，96，6623，199603。程序使用的累加法判断余数是否为0造成了不佳的指令执行数。

0.4 实验总结与思考：

- 本次实验主要通过C程序与LC3程序的转换使我了解了高级语言用一句话实现的功能在底层的一种具体实现方式以及在LC3中如何实现子程序的调用，在实践中关注“现场保留”等概念。
- 通过本次实验，我进一步熟悉了子程序的使用（调用和返回），以及将C程序与LC3程序相关联。