

HW6

T1

如下所示的操作会在栈上执行

```
PUSH A, PUSH B, POP, PUSH C, PUSH D, POP, PUSH E, POP, POP, PUSH F
```

- (a) PUSH F 后，栈中保存什么内容？
- (b) 在何时，栈中拥有最多数量的元素？
- (c) 除了栈的数据结构之外，还有一种名为队列的数据结构，它保证元素的先入先出（FIFO），如果我们把题目中的数据结构修改为队列，重新回答上述问题。

T2

两个学生在作业中编写了中断服务例程。这两个例程的功能完全相同，但第一个学生错误的在例程结尾使用 `RET` 返回，而第二个学生正确的使用了 `RTI`，由于这个错误，第一个学生的程序会出现三个问题，请描述其中的任意两个问题。

T3

- (a) 如果程序不检查 `KBSR` 直接读取 `KBDR` 会发生什么？
- (b) 如果键盘硬件不检查 `KBSR` 直接写 `KBDR` 会发生什么？
- (c) 如果显示硬件不检查 `DSR`，直接写 `DDR` 会发生什么？

T4

描述下述程序的功能：

```
1      .ORIG x3000
2      AND R0, R0, #0
3      ADD R2, R0, #10
4      LD R1, MASK
5      LD R3, PTR1
6  LOOP LDR R4, R3, #0
7      AND R4, R4, R1
8      BRz NEXT
9      ADD R0, R0, #1
10     NEXT ADD R3, R3, #1
11     ADD R2, R2, #-1
12     BRp LOOP
13     STI R0, PTR2
14     HALT
```

```

15 MASK .FILL x8000
16 PTR1 .FILL x4000
17 PTR2 .FILL x5000
18 .END

```

T5

下面的代码实现了逆序输出字符串，请填写空白行：

```

1 .ORIG x3000
2 LEA R6, STACKBASE
3 LEA R0, PROMPT
4 TRAP x22
5 AND R1, R1, #0
6 LOOP TRAP x20
7 TRAP x21
8 ADD R3, R0, #-10
9 BRz INPUTDONE
10 JSR PUSH
11 ADD R1, R1, #1
12 BRnzp LOOP
13 INPUTDONE ADD R1, R1, #0
14 BRz DONE
15 LOOP2 ---
16 TRAP x21
17 ADD R1, R1, #-1
18 BRp LOOP2
19 DONE TRAP x25
20
21 PUSH ---
22 ---
23 ---
24 POP LDR R0, R6, #-2
25 ADD R6, R6, #2
26 RET
27 PROMPT .STRINGZ "Please enter a sentence:"
28 STACKSPAC .BLKW #50
29 STACKBASE .FILL #0
30 .END

```

- 完成 PUSH 处的子函数
- 代码第15行的指令应该是什么？并回答 R1 在代码中扮演的作用是什么？

T6

在本题中，我们修改 JSR / JSRR 和 RET 指令在 LC-3 中的工作方式

JSR / JSRR 将不再使用 R7 存储返回地址，而是将返回地址（PC）压入栈中。相应的，RET 也会从栈顶弹出返回地址并加载到 PC 中。

在内存中，有十个子函数用于输出十个数字。下面的代码展示了输出字符 5 的过程：

```

1 Print5 LD R0 Lable5
2      OUT
3      RET
4 Lable5 .FILL x35

```

因为每一个子函数需要使用 4 个内存地址，每一个子函数的地址会比上一个子函数大4。子函数 Print0 的内存起始地址是 x5000。因此，子函数 Print1 的起始地址是 x5004，子函数 Print2 的起始地址是 x5008，以此类推。

一个学生希望以在主程序中不使用 JSR / JSRR 指令和 IO 中断的方式输出数字 "306"。

任务： 为了实现程序输出 "306" 然后停机的功能，你需要在补全主程序代码中的空白位置。

提示： 如果 RET 指令在之前没有 JSR 指令运行的情况下执行，会发生什么？

```

1      .ORIG x3000
2      LD R6, -----
3      LD R0, -----
4      ADD -----
5      STR R0, R6, #0
6      LD R0, -----
7      ADD -----
8      STR R0, R6, #0
9      LD R0, -----
10     ADD -----
11     STR R0, R6, #0
12     LD R0, -----
13     ADD -----
14     STR R0, R6, #0
15     RET
16     -----
17     -----
18     -----
19     -----
20     HALT
21     SP_INIT .FILL xFE00
22     .END

```

T7

下面的代码完成了从源地址到目标地址拷贝指定大小的内存值的过程：

```

1      .ORIG x300
2      LD R2, DST
3      LD R1, SRC
4      LD R0, SIZE
5 LOOP BRz EXIT
6      ADD R0, R0, #-1
7      ADD R4, R1, R0
8      LDR R5, R4, #0
9      ADD R4, R2, R0
10     STR R5, R4, #0
11     ADD R0, R0, #0
12     BR LOOP

```

```
13  EXIT  HALT
14  SRC   .FILL x4000
15  DST   .FILL x5000
16  SIZE  .FILL x10
17                .END
```

在回答问题的过程中，不要考虑 HALT 指令

- (a) 在运行期间，程序会执行多少条指令 (instructions) ?
- (b) 假设一次内存操作需要 10 个时钟周期 (cycles) ，程序运行需要多少个时钟周期?