## Worksheet - Programming with Stack

**Exercise 1:**

In this exercise, we will write a similar program that reads the memory contents and prints out in reverse order (but not changing the original memory contents). The starting and ending address is stored in R1 and R2.

* + Use PUSH and POP subroutines. Assume the subroutines are provided in the code.
  + You do not have to check the overflow condition.
  + Use the underflow detection (R5) by POP to break LOOP\_POP.
  + *Example*

|  |  |
| --- | --- |
| Address | Value |
| x4000 (starting addr) | x0 |
| x4001 | x1 |
| x4002 | x2 |
| x4003  (ending addr) | x3 |

Result: 3210

|  |
| --- |
| .ORIG x3000  LD R1, ADDR\_A  LD R2, ADDR\_B  LOOP  ;push the item pointed by R1  (1) ;  (2) ;  ADD R1, R1, #1 ;increment starting ptr  ;compare starting ptr and ending ptr  ADD R3, R1, #0  (3) ;  (4) ;  (5) ;  (6)BR LOOP  LOOP\_POP  ;after pushing all items, pop and print them  (7) ;  (8) ;  BRp EXIT  OUT  BRnzp LOOP\_POP  EXIT  HALT  ;IN:R0, OUT:R5 (0-success, 1-fail/overflow)  ;R3: STACK\_END R6: STACK\_TOP  ;  PUSH  ;code omitted  ;OUT: R0, OUT R5 (0-success, 1-fail/underflow)  ;R3 STACK\_START R6: STACK\_TOP  ;  POP  ;code omitted |