For this question, you are going to write a program that checks whether a given word is a palindrome. Your program will have to use a stack to complete this problem. The stack implementation is given.

Details

n = total number of character of given word, it will be stored at x5000

h = floor(total number divide 2), it will be stored at x5001

The given word will be stored in memory starting from x5002 and ends with a NULL character. The stack starts at x4000 and ends at x3FF0. You are given 2 subroutines: PUSH and POP. The PUSH subroutine takes the number from R0 and pushes it onto the stack. If PUSH is successful, R5 will return 0. If PUSH failed, meaning there is an overflow, R5 will return 1. Similarly, the POP subroutine pops the first item on the top of the stack and stores it in R0. If POP is successful, R5 will return 0. If POP failed, meaning there is an underflow, R5 will return 1.

Algorithm

1. Check corner cases. If the given input is empty, print NOT_PA_MSG. If the given input is only 1 character long, print IS_PA_MSG.
2. PUSH first half of the input onto the stack. If the stack overflowed, print out the ERROR_MSG and HALT the program. If n is odd, make sure you skip the middle character.
3. POP one character at a time to compare with rest of the input. If they are not the same, print NOT_PA_MSG. If the input is a palindrome, then print IS_PA_MSG.

Please write your code in stack.asm. You are given 5 different inputs, and you are welcome to modify the input files to test your code.