

# Week 4 - Assembly Assignment

Write an assembly program to find whether a given string is a palindrome or not, using stack operations

Write an assembly program to search a given number in an array

(i)

```
.data
string: .byte 109, 97, 100, 97, 109 # ASCII values for "madam"
len: .word 5 # Length of string
result: .word 0 # Result (1=palindrome, 0=not palindrome)
.text
.globl _start
_start:
la x5, string # Load base address of string
lw x6, len # Load length of string
li x7, 0 # Left index (i)
addi x6, x6, -1 # Right index (j = length - 1)
loop:
bge x7, x6, palindrome # If left >= right, it's a palindrome
# Load characters at left and right indices
add x8, x5, x7 # Address of left char
add x9, x5, x6 # Address of right char
lb x8, 0(x8) # Load left char
lb x9, 0(x9) # Load right char
bne x8, x9, not_palindrome # If chars don't match, not a palindrome
addi x7, x7, 1 # i++
addi x6, x6, -1 # j--
j loop
palindrome:
li x8, 1 # Set result to 1
j store_result
not_palindrome:
li x8, 0 # Set result to 0
store_result:
la x5, result # Load address of result
sw x8, 0(x5) # Store the result
```

(ii) Write an assembly program to search a given number in an array

```
.data
a: .word 0x01223113, 0x45362727, 0x33434343, 0x11211212
b: .word 0x45362727

.text
la x10, a #address of array
la x11, b #address of the no to searched

lw x16, 0(x11) #no to be searched
```

```
addi x20,x0,4 #no of loops
addi x27,x0,0
```

```
loop: lw x12,0(x10) #loading the values of the array
      beq x12,x16,exit #checking if no is present in array
      addi x27,x27,1 #incrementing the loop
      beq x27,x20,exit2 #checking for the loop iteration i<4
      addi x10,x10,4 #incrementing the array address
      beq x0,x0,loop
```

```
exit: addi x25,x0,1 #if x25 is 1 the no is present
exit2: nop
```