**Assembly Sec C Lab 5 Fall 2017**

**TASK 1**

You need to perform bit by bit comparison of two words. If the two words are equal then set carry flag otherwise, unset carry flag, at the end of program value of two words should retain.

**TASK 2**

Write an assembly program that checks **in binary** whether a 16-bit number is palindrome or not. Move 1 in dx register if it is a palindrome else move 0 in dx register. Palindrome is a number which reads the same backward or forward.

For example

0xA425 is a palindrome.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 1 |

**TASK 3**

AX contains a number between 0-15. Write code to complement the corresponding bit in BX. For example if AX contains 6; complement the 6th bit of BX 8. Let BX contain 0.

**TASK 4**

AX contains a non-zero number. Count the number of ones in it and store the result back in AX. Repeat the process on the result (AX) until AX contains one. Calculate in BX the number of iterations it took to make AX one. For example BX should contain 2 in the following case:

AX = 1100 0101 1010 0011 (input – 8 ones)

AX = 0000 0000 0000 1000 (after first iteration – 1 one)

AX = 0000 0000 0000 0001 (after second iteration – 1 one) STOP