

Lab Assignment- 8
Practice Lab
Week 13: 11th Nov -17th Nov, 2022

Lab 9 (8086 Simulator): Array and Strings

1. [CO3] WAP in 8086 to read the string from the user using keyboard interrupt and store the string in memory location DS: 2000H.
2. [CO3] Write an assembly program for extracting the vowels from the string “JIIT IS A UNIVERSITY” and display them. Assume the string is located at some memory location.
3. [CO3] WAP in 8086 to read the string from the memory location and count the number of the vowels in the string and print each count on the screen.

Example: Input: JIIT IS A UNIVERSITY

Output: I:5

A:1

E:1

U:1

4. [CO3] WAP in 8086 to compare two string stored in the two different memory locations.
5. [CO3] WAP in 8086 Assembly to read a sample string as given below and store in Data segment.
Enter the word X to be searched in the inputted String. Find the position as word count of Word X in the String. Send the resultant position as Word Count to be printed on screen.

Input: Welcome JIIT Noida India

Word X: India

Output: Forth

Word X: JIIT

Output: First

6. [CO3] WAP in 8086 Assembly to read a sample string as given below and store in Data segment.
 - (i) Reverse the case and send the resultant string to be printed on screen.
- (ii) Reverse the inputted string and print on screen.

Input: Welcome JIIT

Output: wELCOME jiiT

Input: Welcome JIIT

Output: JIIT Welcome

7. [CO3] 8-bits numbers are stored from location 1FF0h to 1FF9h. Check numbers to find out number of 1's are even or odd in each number, if number of 1's are odd make them even by setting any bit which is 0. And store resulted array of bytes 2990h onwards.

8. **[CO3]** 8-bits numbers are stored from location 1550h to 1559h. Form a new number by complementing D6 and D5 bits of each number and store 2120h onwards.
Example: 44h->20h.

9. **[CO3]** Take a 8-bits (b7, b6, b5, b4, b3, b2, b1, b0) number from memory location 2030h. Perform ex-OR operation on b6 and b1. Write following on 2032h accordingly:

b6	b1	2032h
0	0	'S'
1	1	'N'
1	0	'N'
0	1	'N'

10. **[CO3]** Take a 8-bits (b7, b6, b5, b4, b3, b2, b1, b0) number from memory location 2000h. Swap b6, b5 bits with b2, b1 and store resulted number 2001.

e.g

1000 1111-> 1110 1001