National University of Computer and Emerging Sciences

Laboratory Manual

for

Computer Organization and Assembly Language Programming

(EL 213)

Course Instructor	Ms. Aatira Anum
Lab Instructor(s)	M. Salman Mubarik Rasaal Ahmad
Section	H
Semester	Fall 2023

Department of Computer Science

FAST-NU, Lahore, Pakistan

Objectives

After performing this lab, students shall be able to:

- ✓ Subroutines
- ✓ Stack
- ✓ Local Variables

Exercise 1: Write a subroutine that takes the address of an array of integers, the size of the array, the direction of rotation of the array, and the number of rotations as parameters, and rotates the array left or right the number of times specified. The type of array is ‘word’. Do not use registers for passing arguments to the sub-routine.

1	Address of array	
2	Size of array	A positive integer
3	Direction of Rotation	0 or 1 0= rotate left 1=rotate right
4	Number of rotations	Any positive integer

For instance: arr1: 1, 2, 3, 4

After single shifting left: 2, 3, 4, 1

After single shifting right: 4, 1, 2, 3

After shifting left twice the original array: 3, 4, 1, 2

Exercise 2: Write a subroutine **multiply** that receives two unsigned 8-bit integers and returns **the** product of them back the 16-bit result. Don’t use the mul command to write your own multiply code.

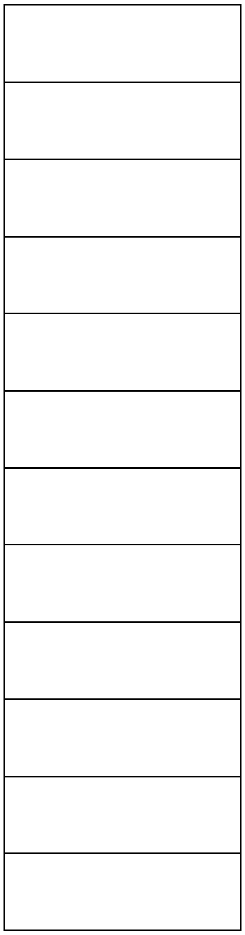
Write a subroutine series that receives two arrays and the size of the array (assume both arrays are the same size), then it computes S = $\sum_{k=0}^{n-1} a(k) * b(k)$, use your multiply subroutine to multiply. then return ‘S’ to your main function. (Assume your ‘S’ never exceeds 16 bits and is a local variable).

Write the main program that passes two arrays and size to a series subroutine then get the ‘S’ and save in CX.

Do not use registers for passing arguments to the sub-routine. You may use a register for returning value from a sub-routine.

Exercise 3: Make an array of 0x80 bytes and treat it as one 0x400 bits. Write a function myalloc that takes one argument, the number of bits. It finds that many consecutive zero bits in an array, make them one.

<u>Exercise 4:</u> A function takes three parameters P1, P2, and P3, returns two values Output1, and Output2 declares one local variable (local1) and saves AX, BX, and CX registers. What will the stack configuration be for this function after pushing all these variables and registers? Fill in the SP given stack. Also, specify where BP and SP should be pointing.



STACK