

COAL Lab (EL2003)

Date: 14-Oct-2024

Course Instructor(s)

UBAID ULLAH

Lab Mid Exam (A)

Total Time: 90 minutes

Total Marks: 20

Total Questions: 03

Semester: FL-2024

Campus: Karachi

Dept: BSE-3A

Submission Instructions:

- Create a word file and paste your all solution code and screen shot of your output, please do not take or attach screen shot of the code.
- Submission is via a client software so open the application present on the Desktop.
- Enter your username as 24K-xxxx and its assigned password (Default is Fast1234).
- Submission is timed so after the time no submission will be accepted.

Student Name

Roll No

Section

Student Signature

LO # 2: Understand and hands-on Data Definitions, General Purpose Registers, moving data to registers, addition, subtraction and basic arithmetic operations in assembly language with implementation.

1.

- You are tasked with developing a small program for a simple arithmetic processor in assembly language. This processor needs to handle a series of operations on two signed 8-bit integers and one unsigned 16-bit integer. The operations should be carried out step by step, modifying the values and storing results in registers. The program should execute the following sequence of tasks: [3.5 marks]

- The processor starts by reading two signed 8-bit integers (signedByte1 and signedByte2) and one unsigned 16-bit integer (unsignedWord).
- It then needs to load the values into registers. The signed 8-bit integers should be sign-extended to 32-bit registers, while the unsigned 16-bit integer should be zero-extended to fit into a 32-bit register.
- The processor must swap the values of the two signed integers. After swapping, each register holds the value of the other signed integer.
- Once swapped, the processor should increment the first signed integer and decrement the second signed integer.
- The processor should then add the two signed integers together, storing the result in a register.
- After that, the second signed integer should be subtracted from the first signed integer, updating the result.
- The result from the previous step (first signed integer) is then multiplied by the unsigned integer, using signed multiplication.
- The final result is negated, ensuring the opposite of the computed value is stored in a register.
- Finally, the processor should output the computed result.

Implement the following formula: [1.5 marks]

$$(a + b)^3 = a^3 + b^3 + 3a^2b + 3ab^2$$

National University of Computer and Emerging Sciences

CLO # 2: Gain hands on experience in writing code that provides the use of arrays and accessing the array elements using indexed addressing with scale factors.

Q2. [5 marks]

You are tasked with creating a **Sales Tracking System** for a small store that sells three different products. The store owner wants to input the sales amounts for each product at the end of the day and calculate the total sales. The system should store the sales amounts for the three products in an array, using indexed addressing with scale factors to ensure proper access to the array elements. The user will be prompted to enter the sales amount for each product one by one. Each sales amount will be stored in the corresponding index of the array. After all inputs are received, the program will calculate the total sales by summing the amounts in the array. This calculation will also use indexed addressing with scale factors to access the elements. The total sales will then be displayed to the user. The program should not use loops but instead rely on direct access to array elements using the scale factor. Additionally, the system must output a formatted message showing the total sales amount after the calculation is completed.

CLO # 2: Understand and implement code utilize `jmp`, `cmp` instructions, for decision making, loops or iteration and arrays for data storage and manipulation, demonstrating efficient management of elements in array.

3.

- a. Convert the given code C-code in assembly language by using only `jmp` instruction. [4 marks]

```
13 if (x>0) {  
    14     if (x==y) {  
        15         x=x*y;  
        16         y=y-2;  
    17     }  
    18     Else {  
        19         x=x*x;  
        20         y=y-1;  
    21     }  
22 }  
23 else {  
24     printf("wrong input");  
25 }
```

- b. You have been tasked with creating a student score tracker that allows teachers to enter test scores for 5 students along with their 3 digit student id in a classroom. After all scores are entered, the program should calculate the total score, and determine which student had the highest, lowest score (print student_ID of the students along with marks). In this context, describe how the program handles user input, updates the sum and maximum, minimum score. You can use (direct, indirect or indexed operands and you can use loop). [6 marks]