National University of Computer and Emerging Sciences
Karachi Campus

# Computer Organization and Assembly language

# Sessional-II Exam

Total Time (Hrs): 1
Total Marks: 30
Total Questions: 3

Date: November 05th 2024

**Course Instructor(s)**

Mr. Shoaib Rauf, Mr.Kashan Hussain, Mr. Aashir Mahboob, Ms. Atiya Jokhio, Mr. M.Kariz Kamal, Mr. M.Usman, Mr. Nauraiz Subhan

______________ ______________ ______________________

Roll No Section Student Signature

**Do not write below this line**

## INSTRUCTION: Attempt all the questions in-order.

*CLO # 2:* Identify use of processor register and addressing modes in various instructions for solving arithmetic data transfer and conditional processing

**Q1:** You have the two collections of number that are Multiplicand and Multiplier respectively. You need to multiply each element of Multiplicand to its corresponding Multiplier and store the result in Product array. You must use loop instruction to solve this problem **[Marks: 10]**

```
Multiplicand  BYTE   31h , 6Bh , 0Ch, 11h,  2Fh
Multiplier    WORD   1Ch, 90h, 3Ah, 16h, 1Eh
Product       DWORD  5 DUP(0)
```

*CLO # 2:* Identify use of processor register and addressing modes in various instructions for solving arithmetic data transfer and conditional processing

**Q2:** Convert the following code into assembly language. Keep in mind that all the comparison instruction must be implemented through eflag register status flag (carry flag and zero flag).

**[Marks: 10]**

```
int array[] = {3,29,101,65,53};  int size, temp;
for (int step = 0; step < size - 1; step++)
 {
        int min_idx = step;
        for (int i = step + 1; i < size; i++)
        {
            if (array[i] < array[min_idx])
                 min_idx = i;
        }
        temp = array[min_idx];
        array[min_idx] = array[step];
        array[step] = temp;
 }
```

*CLO # 4:* **Analyse basic assembly program of x86 and RISC ISAs**

**Q3: Solve the following question.** **[Marks: 5+5 = 10]**

I. Write an assembly language program that converts the following 32-bit hexadecimal number **12438765h** to **87654321h** using shift and rotate instructions.

II. Given that EAX = 0Eh, ECX = 17h, EDX = 02h, and ESP = 0000 011Eh, draw out the run-time stack (diagrams), with addresses after each numbered (a, b and c) instruction. No points will be awarded if addresses are found missing/wrong.

```
Main PROC
        SUB  AL, 1
        INC  DH
        PUSH EAX        ;a= _______
        SHL  CL,2
        PUSH ECX        ;b= _______
        ROR  DL,1
        PUSH EDX        ;c= _______

        POP  EDX
        POP  ECX
        POP  EAX
Main ENDP
```

================== **Good Luck** ==================