

Course Instructor(s)
Muhammad Nadeem Ghouri

Total Marks: 20
Total Questions: 03

Semester: Fall-2024
Campus: Karachi
Department: Computer Science

mission Instructions:

- You must comment your student ID on top of each file. (Line#1 of your code).
Name the file for each question according to Roll_No e.g. k23-xxxx_Q1.c, k23-xxxx_Q2.c etc.
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
CLO # 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.			
Q1.			[2.5 + 2.5 Marks]

- (a) Write program that calculates the following expression, using registers: $A = (A + B) - (C - D)$.
Assign integer values to the EBX, ECX, EDX registers.
Note: Consider the values of A, B, C and D according your roll number. So, for example your roll number K23-5678 takes the values of A = 5, B = 6, C = 7 and D = 8.
- (b) Consider your roll number. For example, the roll number 23K-1234. Write an assembly program that creates an array to store the characters of your roll number and then counts the number of alphabetic characters (both uppercase and lowercase) in your roll number. Display the count as the output.

CLO # 2: Gain hands on experience in writing code that provides the use of arrays and accessing the array elements using indexed addressing.	
Q2.	[5 Marks]

Using a loop and indexed addressing, write a code that rotates the numbers of 32-bit integer array Forward one position. The value at the end of the array must wrap around to the first position. For example, the array [50, 60, 70, 80] would be transformed into [80, 70, 60, 50].

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

Q3.

[05 + 05 Marks]

- (a) Write an assembly language program that takes an array of integers as input from the user and checks whether all the numbers in the array are odd. If all numbers are odd, the program should display 1 (indicating "all odd"); otherwise, it should display 0. The array size should also be entered by the user.

.data

sizeMsg BYTE "Enter the size of the array: ", 0

elementMsg BYTE "Enter array elements: ", 0

resultMsg BYTE "Result: ", 0

SAMPLE INPUT-OUTPUT:

```
Enter the size of the array: 4
Enter array elements: 3 5 7 9
Result (1 = All Odd, 0 = Not All Odd): 1
```

```
Enter the size of the array: 5
Enter array elements: 2 3 4 5 6
Result (1 = All Odd, 0 = Not All Odd): 0
```

- (b) Write an assembly language program that takes the marks of multiple students across three subjects and calculates the number of students who passed all subjects. A student is considered to have passed if their mark in all subjects is greater than or equal to 50.

.data

subject1 DWORD 70, 45, 60, 55, 80

subject2 DWORD 65, 55, 80, 50, 70

subject3 DWORD 80, 65, 50, 60, 90

SAMPLE EXPECTED OUTPUT: 4

Good Luck!