

1.) Perform 3 examples in each of the addressing modes in MCU 8051 IDE

```
ORG 0  
  
; Immediate  
MOV A, #25h  
MOV R0, #69H  
MOV R1, #44H
```

The screenshot shows the MCU 8051 IDE interface. On the left, a memory dump table displays addresses from 00 to 40. Address 00 is highlighted in green, and its value 69 is highlighted in orange. Address 01 has a value of 44, also highlighted in orange. All other memory locations contain 00. On the right, the register status panel shows the following values:

HEX	DEC	BIN	OCT	CHAR
A: 25	37	00100101	45	%
B: 00	0	00000000	0	

Below the register status, the PSW (Program Status Word) is shown as: C AC F0 RS1 RS0 OV - P. At the bottom, the register values for R7 through R0 are displayed in a row: R7 (00), R6 (00), R5 (00), R4 (00), R3 (00), R2 (00), R1 (44), and R0 (69).

```
; Register  
MOV A, R0  
MOV A, R1  
MOV B, A
```

The screenshot shows the MCU 8051 IDE interface after executing register addressing mode instructions. The memory dump table on the left remains the same as in the previous screenshot, with address 00 containing 69 and address 01 containing 44. The register status panel on the right now shows:

HEX	DEC	BIN	OCT	CHAR
A: 44	68	01000100	104	D
B: 44	68	01000100	104	D

The PSW remains: C AC F0 RS1 RS0 OV - P. The register values at the bottom are still: R7 (00), R6 (00), R5 (00), R4 (00), R3 (00), R2 (00), R1 (44), and R0 (69).

```
; Direct
MOV 02H, 00H
MOV 03H, 01H
MOV 01H, 05H
```

	x0	x1	x2	x3	x4	x5	x6	x7
00	69	00	69	44	00	00	00	00
08	00	00	00	00	00	00	00	00
10	00	00	00	00	00	00	00	00
18	00	00	00	00	00	00	00	00
20	00	00	00	00	00	00	00	00
28	00	00	00	00	00	00	00	00
30	00	00	00	00	00	00	00	00
38	00	00	00	00	00	00	00	00
40	00	00	00	00	00	00	00	00

```
; Indirect
MOV R1, #30H
MOV 30H, #69H
XCH A, @R1
MOV R0, #42H
MOV 42H, #44H
ADD A, @R0
MOV @R1, A
```

The screenshot shows the Proteus simulator interface. The top menu bar includes "Simulator", "C variables", "IO Ports", "Messages", and "Notes".

The "C variables" window displays a memory dump for variable `x0`. It shows values in hexadecimal, decimal, octal, and character formats across multiple rows.

	<code>x0</code>	<code>x1</code>	<code>x2</code>	<code>x3</code>	<code>x4</code>	<code>x5</code>	<code>x6</code>	<code>x7</code>
00	42	30	69	0x00				
08	00	00	00	DEC	HEX	OCT		
10	00	00	00	66	42	102		
18	00	00	00	BIN	CHAR			
20	00	00	00					
28	00	00	00	01000010	B			
30	AD	00	00	00	00	00	00	00
38	00	00	00	00	00	00	00	00
40	00	00	44	00	00	00	00	00

The "IO Ports" window displays the PSW (Program Status Word) and registers R7 through R0.

PSW: C AC F0 RS1 RS0 OV - P

R7	R6	R5	R4	R3	R2	R1	R0
00	00	00	00	44	69	30	42

- 2.) Initialize your values in registers R0 to R7 with different values, then sort manually using the mov instruction and store it in the R0 to R7 registers.

```

1  ORG 0
2
3  MOV R0, #5
4  MOV R1, #4
5  MOV R2, #3
6  MOV R3, #2
7  MOV R4, #1
8  MOV R5, #0
9  MOV R6, #6
10 MOV R7, #7
11
12 MOV A, R0
13 MOV B, R5
14
15 MOV R0, B
16 MOV R5, A
17
18 MOV A, R1
19 MOV B, R4
20
21 MOV R1, B
22 MOV R4, A
23
24 MOV A, R2
25 MOV B, R3
26
27 MOV R2, B
28 MOV R3, A
29
30 END

```

3.) Swap two values in the data memories 20H and 30H.

```

1 ORG 0
2
3 MOV 20H, #20H
4 MOV 30H, #30H
5 MOV R0, 20H
6 MOV 20H, 30H
7 MOV 30H, R0
8
9 END

```

	x0	x1	x2	x3	x4	x
00	20	00	00	00	00	0
08	00	00	00	00	00	0
10	00	00	00	00	00	0
18	00	00	00	00	00	0
20	30	00	00	00	00	0
28	00	00	00	00	00	0
30	20	00	00	00	00	0
38	00	00	00	00	00	0