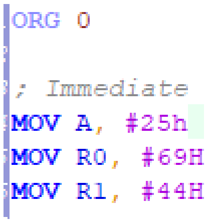
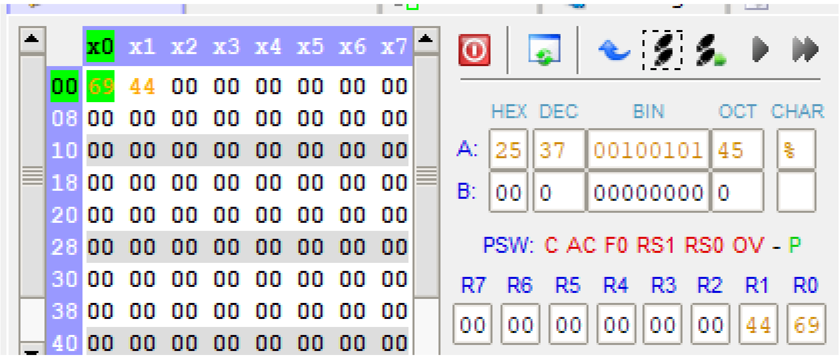
1.) Perform 3 examples in each of the addressing modes in MCU 8051 IDE

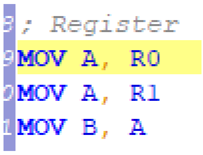


```
ORG 0

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

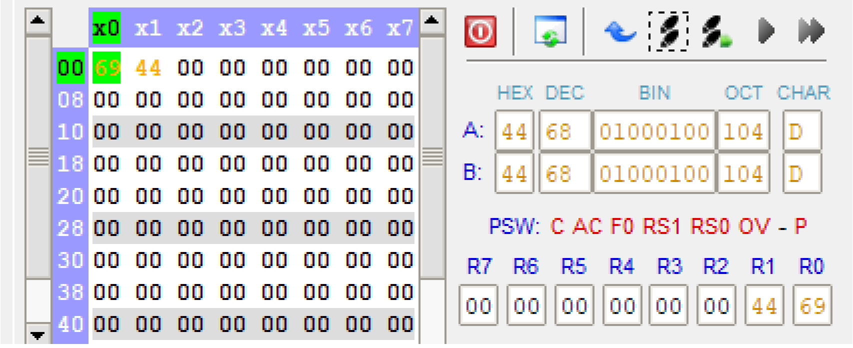


```
   x0 x1 x2 x3 x4 x5 x6 x7
00 69 44 00 00 00 00 00 00
08 00 00 00 00 00 00 00 00
10 00 00 00 00 00 00 00 00
18 00 00 00 00 00 00 00 00
20 00 00 00 00 00 00 00 00
28 00 00 00 00 00 00 00 00
30 00 00 00 00 00 00 00 00
38 00 00 00 00 00 00 00 00
40 00 00 00 00 00 00 00 00
```

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

PSW: C AC F0 RS1 RS0 OV - P

R7 R6 R5 R4 R3 R2 R1 R0
00 00 00 00 00 00 44 69
```

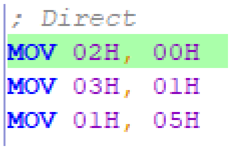


```
; Register
MOV A, R0
MOV A, R1
MOV B, A
```

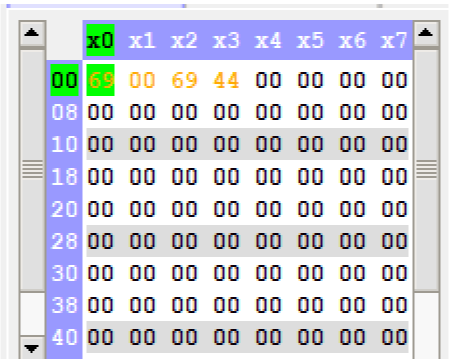


```
   x0 x1 x2 x3 x4 x5 x6 x7
00 69 44 00 00 00 00 00 00
08 00 00 00 00 00 00 00 00
10 00 00 00 00 00 00 00 00
18 00 00 00 00 00 00 00 00
20 00 00 00 00 00 00 00 00
28 00 00 00 00 00 00 00 00
30 00 00 00 00 00 00 00 00
38 00 00 00 00 00 00 00 00
40 00 00 00 00 00 00 00 00
```

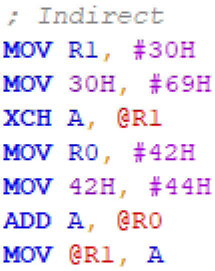
```
    HEX DEC BIN      OCT CHAR
A:  44  68  01000100 104 D
B:  44  68  01000100 104 D

PSW: C AC F0 RS1 RS0 OV - P

R7 R6 R5 R4 R3 R2 R1 R0
00 00 00 00 00 00 44 69
```



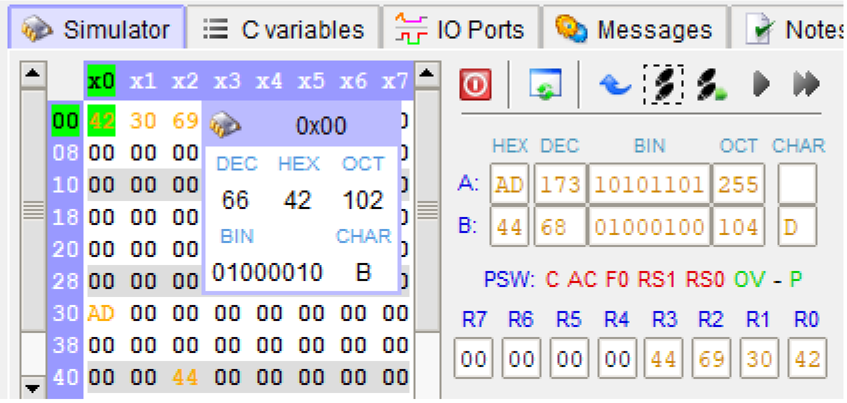
```
; Direct
MOV 02H, 00H
MOV 03H, 01H
MOV 01H, 05H
```



	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
```



	x0	x1	x2	x3	x4	x5	x6	x7
00	42	30	69					
08	00	00	00					
10	00	00	00					
18	00	00	00					
20	00	00	00					
28	00	00	00					
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

0x00

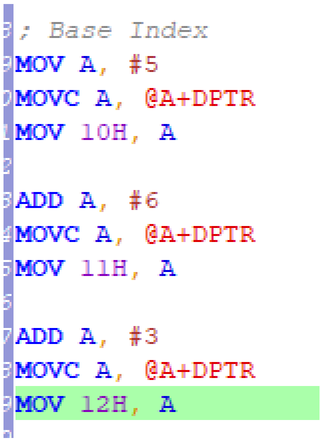
DEC	HEX	OCT
66	42	102

BIN	CHAR
01000010	B

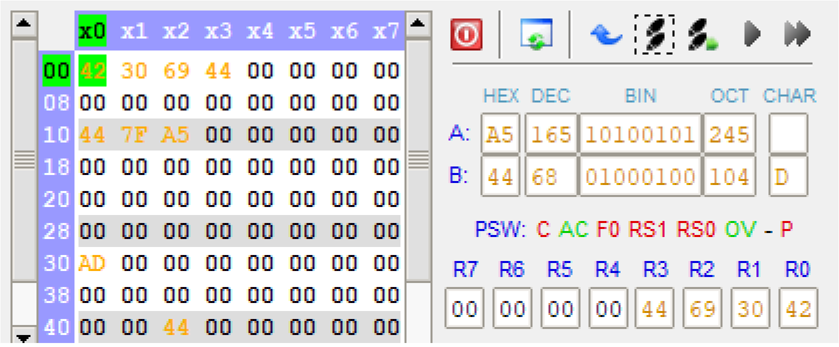
	HEX	DEC	BIN	OCT	CHAR
A:	AD	173	10101101	255	
B:	44	68	01000100	104	D

PSW: C AC F0 RS1 RS0 OV - P

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



```
; Base Index
MOV A, #5
MOVC A, @A+DPTR
MOV 10H, A

ADD A, #6
MOVC A, @A+DPTR
MOV 11H, A

ADD A, #3
MOVC A, @A+DPTR
MOV 12H, A
```



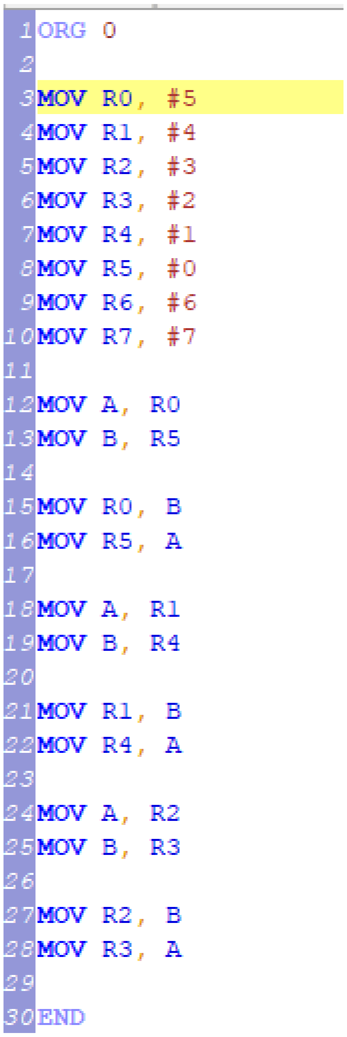
```
    x0 x1 x2 x3 x4 x5 x6 x7
00  42 30 69 44 00 00 00 00
08  00 00 00 00 00 00 00 00
10  44 7F A5 00 00 00 00 00
18  00 00 00 00 00 00 00 00
20  00 00 00 00 00 00 00 00
28  00 00 00 00 00 00 00 00
30  AD 00 00 00 00 00 00 00
38  00 00 00 00 00 00 00 00
40  00 00 44 00 00 00 00 00
```

	HEX	DEC	BIN	OCT	CHAR
A:	A5	165	10100101	245	
B:	44	68	01000100	104	D

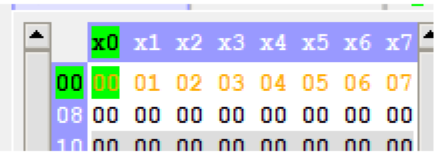
PSW: C AC F0 RS1 RS0 OV - P

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.

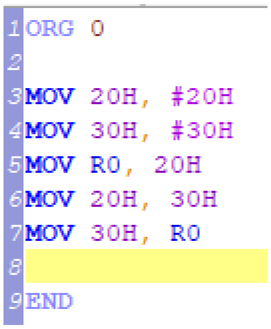


```
ORG 0

MOV R0, #5
MOV R1, #4
MOV R2, #3
MOV R3, #2
MOV R4, #1
MOV R5, #0
MOV R6, #6
MOV R7, #7

MOV A, R0
MOV B, R5

MOV R0, B
MOV R5, A

MOV A, R1
MOV B, R4

MOV R1, B
MOV R4, A

MOV A, R2
MOV B, R3

MOV R2, B
MOV R3, A

END
```

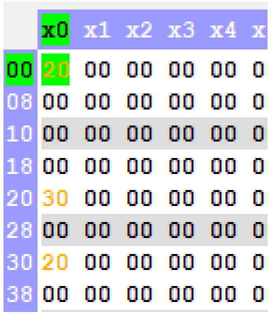


```
   x0 x1 x2 x3 x4 x5 x6 x7
00 00 01 02 03 04 05 06 07
08 00 00 00 00 00 00 00 00
10 00 00 00 00 00 00 00 00
```

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



```
ORG 0

MOV 20H, #20H
MOV 30H, #30H
MOV R0, 20H
MOV 20H, 30H
MOV 30H, R0

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
```