

- Assignment 3
  - Question 1
    - Objective
    - Tool / Experimental setup considered
    - Procedure
    - Program
    - Experimentation
    - Conclusion
  - Question 2
    - Objective
    - Tool / Experimental setup considered
    - Procedure
    - Program
    - Experimentation
    - Conclusion
  - Question 3
    - Objective
    - Tool / Experimental setup considered
    - Program
    - Experimentation
    - Conclusion

## Assignment 3

---

### Question 1

---

#### Objective

Write a fast sub-routine to multiply 9 by 15.

#### Tool / Experimental setup considered

- Used [Jubin's 8085 Simulator](#).

#### Procedure

We can add 15, 9 times to get 9\*15 because multiplication is nothing but repeated addition.

#### Program

```
LXI H,0000H
MVI E,0FH
MVI B,09H
LOOP:
    DAD D ; HL <- HL + DE
    DCR B ; Decrease B (1-byte)
    JNZ LOOP
HLT
```

#### Experimentation

Registers										
Memory										
Devices										
Registers :										
Register	Value	7	6	5	4	3	2	1	0	
Accumulator	00	0	0	0	0	0	0	0	0	
Register B	00	0	0	0	0	0	0	0	0	
Register C	00	0	0	0	0	0	0	0	0	
Register D	00	0	0	0	0	0	0	0	0	
Register E	0F	0	0	0	0	1	1	1	1	
Register H	00	0	0	0	0	0	0	0	0	
Register L	87	1	0	0	0	0	1	1	1	
Memory(M)	00	0	0	0	0	0	0	0	0	
Resister	Value	S	Z	*	AC	*	P	*	CY	
Flag Register	54	0	1	0	1	0	1	0	0	
Type	Value									
Stack Pointer(SP)	0000									
Memory Pointer (HL)	0087									
Program Status Word(PSW)	0054									
Program Counter(PC)	000C									
Clock Cycle Counter	242									
Instruction Counter	31									

## Conclusion

$9 \times 15 = 135$  which in hex is `0x87`. Above we can see that the value in the H-L pair is `0087h`. Hence, our program was able to calculate 9 times 15.

## Question 2

### Objective

Write a subroutine to sort a 5-element byte array (Any algorithm will do).

### Tool / Experimental setup considered

- Used [Jubin's 8085 Simulator](#).

### Procedure

Performing a simple bubble sort, where we compare adjacent elements and swaps them if they are in the wrong order. The pass through the list is repeated until the list is sorted.

### Program

```
# ORG 4000h
# ARR: DB 04,05,01,02,03
# LEN EQU 5

# ORG 0000h

MAIN:
    LXI H,ARR
    MVI D,00
    MOV C,LEN
    DCR C
CHECK:
    MOV A,M
    INX H
    CMP M
    JC NEXT
    JZ NEXT
    CALL SWAP
SWAP:
    MOV B,M
    MOV M,A
    DCX H
```

```
NEXT :
    DCR C
    JNZ CHECK
    MOV A, D
    CPI 01
    JZ MAIN
    HLT
```

Finally we can see in the memory range 4000 to 4004 that the numbers are all sorted, hence our code worked.

## Question 3

---

### Objective

Write a sub-routine to STORE all the registers ( A , F , B , C , D , E , H , L , I , SPL , SPH , PCL , PC , in that order) starting from location MYREGISTERS .

### Tool / Experimental setup considered

- Used [Jubin's 8085 Simulator](#).

### Program

```
# ORG 1000H
# MYREG EQU 2000H
    MVI A,10
    LXI B,1001
    LXI D,2002
    LXI H,3003
    SIM
    LXI SP,F001h
    CALL MAIN
    HLT
```

```
MAIN:
    PUSH H
    PUSH D
    PUSH B
    PUSH PSW
    LXI H,MYREG
    POP D
    CALL STORE
    POP D
    CALL STORE
    POP D
    CALL STORE
    POP D
    CALL STORE
    RIM
    MOV M,A
    INX H
    XCHG
    LXI H,0000
    DAD SP
    XCHG
    INX D
    INX D
    CALL STORE
    XCHG
    XTHL
    XCHG
    CALL STORE
    XCHG
    XTHL
    XCHG
    RET
```

```
STORE:
    MOV M,D
    INX H
    MOV M,E
    INX H
    RET
```

### Experimentation

Registers <b>Memory</b> Devices		
Memory Editor		
Memory Range: 0000 ---- FFFF		
Memory Address	Value	
102B	77	▲
102C	23	≡
102D	EB	
102E	21	
1031	39	
1032	EB	
1033	13	
1034	13	
1035	CD	
1036	42	
1037	10	
1038	EB	
1039	E3	
103A	EB	
103B	CD	
103C	42	
103D	10	
103E	EB	
103F	E3	
1040	EB	
1041	C9	
1042	72	
1043	23	
1044	73	
1045	23	
1046	C9	

## Conclusion

It is storing all the registers in that order mentioned in the question.