

MIT ASSIGNMENT – 6

1. The following block of data is stored in memory locations from 3055H to 305AH. Write a program to transfer the block of data in reverse order at same memory location.

DATA (HEX): 22, A5, B2, 99, 7F, 37

;Program1

```
lxi h,3045h  
lxi d,304Ah  
call reverse  
hlt
```

;Subroutine

```
reverse: mov b,m  
xchg  
mov c,m  
mov m,b  
xchg  
mov m,c  
inx h  
dcx d  
stc  
cmc  
mov a,e  
sub l  
rz  
rc  
jmp reverse
```

Before execution -

Data
Stack
KeyPad
Memory
I/O Ports

Start

Address (Hex)	Address	Data
3045	12357	22
3046	12358	165
3047	12359	178
3048	12360	99
3049	12361	127
304A	12362	37
304B	12363	0
304C	12364	0
304D	12365	0
304E	12366	0
304F	12367	0
3050	12368	0

Line No	Assembler Message
0	Program assembled successfully

After execution -

Data
Stack
KeyPad
Memory
I/O Ports

Start

Address (Hex)	Address	Data
3045	12357	37
3046	12358	127
3047	12359	99
3048	12360	178
3049	12361	165
304A	12362	22
304B	12363	0
304C	12364	0
304D	12365	0
304E	12366	0
304F	12367	0
3050	12368	0

Line No	Assembler Message
0	Program assembled successfully

2. Find the square of the given numbers from memory location 6100H and store the result from memory location 7000H.

;Program2

lxi h,6100h

lxi d,7000h

call square

hlt

;Subroutine

square: mvi a,00h

```

mov c,m
mov b,m
rep: add c
dcr b
jnz rep
stax d
inx d
inx h
mov a,e
cpi 05h
rz
jmp square

```

Data	Stack	KeyPad	Memory	I/O Ports
Start	6100h	OK		
Address (Hex)	Address	Data		
6100	24832	1		
6101	24833	3		
6102	24834	5		
6103	24835	7		
6104	24836	8		
6105	24837	0		
6106	24838	0		
6107	24839	0		
6108	24840	0		
6109	24841	0		
610A	24842	0		
610B	24843	0		
Line No	Assembler Message			
0	Program assembled successfully			

Square of the numbers are -

Data	Stack	KeyPad	Memory	I/O Ports
Start	7000h	OK		
Address (Hex)	Address	Data		
7000	28672	1		
7001	28673	9		
7002	28674	25		
7003	28675	49		
7004	28676	64		
7005	28677	0		
7006	28678	0		
7007	28679	0		
7008	28680	0		
7009	28681	0		
700A	28682	0		
700B	28683	0		
Line No	Assembler Message			
0	Program assembled successfully			

3. WAP to find Factorial of a given number using Call and Subroutine.

;Program3

```
lhld 3000h
mvi d,0
mvi e,0
xchg
mov b,e
start: dcr b
mov a,b
cpi 01
jz end
mov c,b
call factorial
mov d,h
mov e,l
mov b,c
mvi h,0
mvi l,0
jmp start
end: xchg
shld 3002h
hlt
```

;Subroutine

```
factorial: dad d
dcr b
jz rt
call factorial
rt: ret
```

Data	Stack	KeyPad	Memory	I/O Ports
Start 3000h OK				
Address (Hex)	Address	Data		
3000	12288	4		
3001	12289	0		
3002	12290	24		
3003	12291	0		
3004	12292	0		
3005	12293	0		
3006	12294	0		
3007	12295	0		
3008	12296	0		
3009	12297	0		
300A	12298	0		
300B	12299	0		
Line No	Assembler Message			
0	Program assembled successfully			

4. WAP for Fibonacci Series using Call and Subroutine.

;Program4

```
mvi b,01h
mvi c,01h
mvi d,00h
lda 3000h
lxi h,3001h
mvi m,01h
dcr a
jz end
inx h
mvi m,01h
dcr a
jz end
mov e,a
call fibonacci
end: hlt
```

;Subroutine

```
fibonacci: mov a,b
mov b,c
add c
mov c,a
inx h
mov m,a
dcr e
jz rt
call fibonacci
rt: ret
```

Data	Stack	KeyPad	Memory	I/O Ports
Start 3000h OK				
Address (Hex)	Address	Data		
3000	12288	9		
3001	12289	1		
3002	12290	1		
3003	12291	2		
3004	12292	3		
3005	12293	5		
3006	12294	8		
3007	12295	13		
3008	12296	21		
3009	12297	34		
300A	12298	0		
300B	12299	0		
Line No	Assembler Message			
0	Program assembled successfully			

The Fibonacci series starts from address 3001h to 3009h.

5. WAP to find Multiplication of Two 8-Bit Numbers using Call and Subroutine

;Program5

```
lda 3000h
mov b,a
lda 3001h
mov c,a
call multiply
sta 3002h
hlt
```

;Subroutine

```
multiply: dcr b
jz rt
add c
jmp multiply
rt: ret
```

The screenshot shows a software interface for an 8085 assembly simulator. At the top, there are tabs for 'Data', 'Stack', 'Keypad', 'Memory', and 'I/O Ports'. The 'Memory' tab is selected. Below the tabs, there is a 'Start' field containing '3000h' and an 'OK' button. A table displays memory contents with three columns: 'Address (Hex)', 'Address', and 'Data'. The table lists memory locations from 3000 to 300B, with corresponding internal addresses (12288 to 12299) and data values (6, 4, 24, and zeros). At the bottom, a section titled 'Line No' and 'Assembler Message' shows line 0 with the message 'Program assembled successfully'.

Address (Hex)	Address	Data
3000	12288	6
3001	12289	4
3002	12290	24
3003	12291	0
3004	12292	0
3005	12293	0
3006	12294	0
3007	12295	0
3008	12296	0
3009	12297	0
300A	12298	0
300B	12299	0

Line No	Assembler Message
0	Program assembled successfully