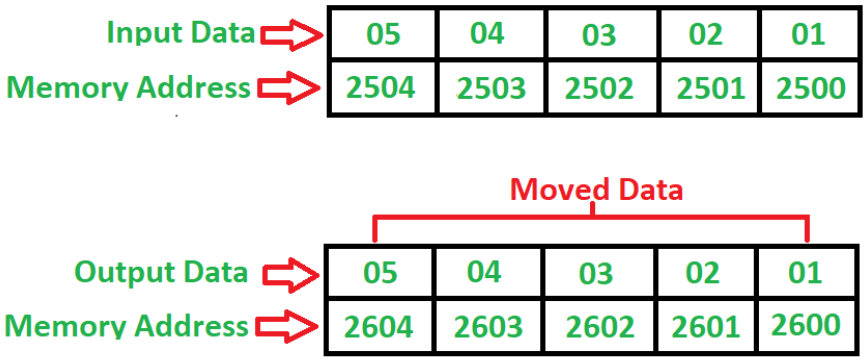
Memory Block

1. Write a program to copy blocks of bits from source location to a destination location.



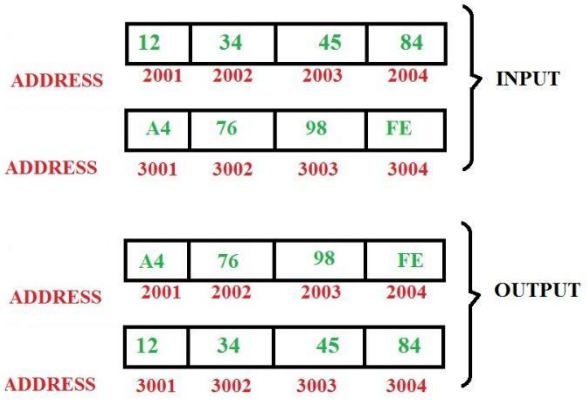
Algrithm:

a. Load register pair H-L with the address 2500H
b. Load register pair D-E with the address 2600H
c. Move the content at memory location into accumulator.
d. Store the content of accumulator into memory pointed by D-E
e. Increment value of register pair H-L and D-E by 1
f. Decrements value of register C by 1
g. If zero flag is not equal to 1, go to step 3.
h. Stop

Program

```
MVI C, 05
LXI H, 0000H
LXI D, 1000H
LOOP: MOV A, M
STAX is used to store A into register pair indirectly (3 Byte instruction)
STAX D
INX H
INX D
DCR C
JNZ LOOP
HLT
```

2. Write an assembly level program to exchange data between two memory location.



Algorithm

1. Take a count equal to 4.
2. Store the starting address of both blocks in 2 different register pairs.
3. Now exchange the contents at the addresses in both register pairs
4. Increment the values of both register pairs.
5. Decrements count by 1.
6. If count is not equal to 0 repeat steps 3 to 5

Program

```
MVI C, 05
LXI H, 0000H
LXI D, 0006H
LOOP: MOV B, M
```

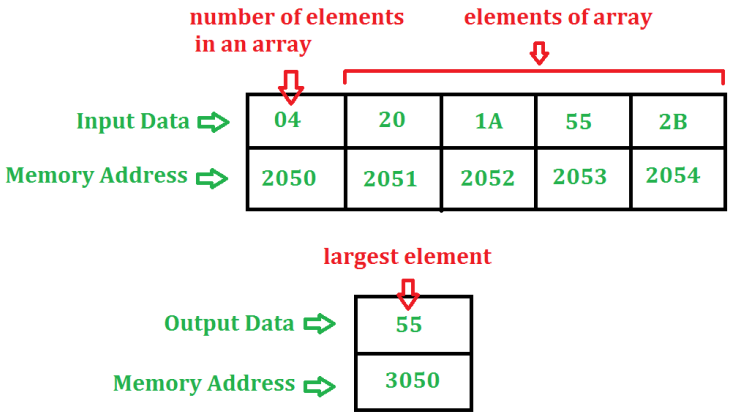
Here Accumulator is loaded with the data stored at address formed by register pair D – E

```
LDAX D
MOV M, A
MOV A, B
```

Stores the content of A (accumulator) in the address formed by register pair D – E.

```
STAX D
INX H
INX D
DCR C
JNZ LOOP
HLT
```

3. Write an Assembly language program to find the largest number in an array.



Algorithm:

1. We are taking first element of array in A
2. Comparing A with other elements of array, if A is smaller then, store that element in A, otherwise compare with next element.
3. The value of A is the answer.

Program for largest:

```
LXI H, 0000H
MOV C, M
INX H
MOV B, M
DCR C
LOOP: INX H
MOV A, M
CMP B
JC SKIP
MOV B, A
SKIP: DCR C
JNZ LOOP
LXI H, 000AH
MOV M, B
HLT
```

Program for smallest:

```
LXI H, 0000H
MOV C, M
INX H
MOV B, M
DCR C
LOOP: INX H
MOV A, M
CMP B
JNC SKIP
MOV B, A
SKIP: DCR C
JNZ LOOP
LXI H, 000AH
MOV M, B
HLT
```