

Lab-1

Obj-1:

```
MOV AX,5225H
MOV CX,AX
ADD AX,4324H
MOV [5000H],AX
MOV AX,CX
SUB AX,4324H
MOV [5002H],AX
MOV AX,CX
MOV BX,4324H
MUL BX
MOV [5004H],AX
MOV [5006H],DX
MOV DX, 0000H
MOV AX,CX
DIV BX
MOV [5008H],AX
MOV [500AH],DX
HLT
```

RESULT OF MEMORY AND Flag register

0100:5000

update

☒ table ☐ list

0100:5000	49	95	01	0F	34	3C	8B	15-25	52	F1	15	00	00	00	00
0100:5010	00	00	00	00	00	00	00	00-00	00	00	00	00	00	00	00
0100:5020	00	00	00	00	00	00	00	00-00	00	00	00	00	00	00	00
0100:5030	00	00	00	00	00	00	00	00-00	00	00	00	00	00	00	00
0100:5040	00	00	00	00	00	00	00	00-00	00	00	00	00	00	00	00
0100:5050	00	00	00	00	00	00	00	00-00	00	00	00	00	00	00	00
0100:5060	00	00	00	00	00	00	00	00-00	00	00	00	00	00	00	00
0100:5070	00	00	00	00	00	00	00	00-00	00	00	00	00	00	00	00

flags

CF

1

ZF

0

SF

0

OF

1

PF

0

AF

0

IF

1

DF

0

Objectives -2:

**Swapping of nibble of data1 and Swapping of nibble of data1
and Y= (data1 and data2) or (data1 xor data2)**

;swapping of nibble

mov si, 1000h

mov al,[si] ;1st data stores in al

mov cl,al

inc si

mov bl,[si] ;2nd data stores in bl

mov dl,al

shr al,04

shl dl,04

or al,dl

inc si

mov [si],al

;Computation of Y

mov al,cl

and al,bl

mov dl,al

xor cl,bl

or al,cl

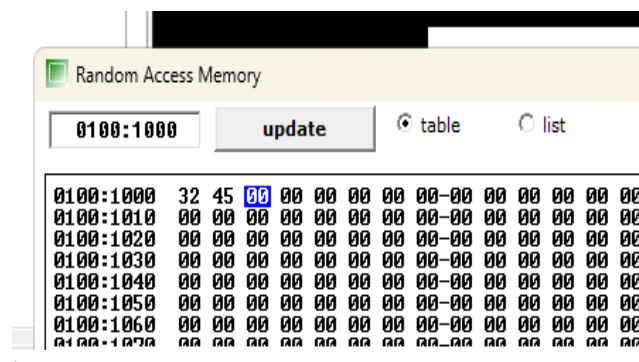
inc si

mov [si],al

hlt

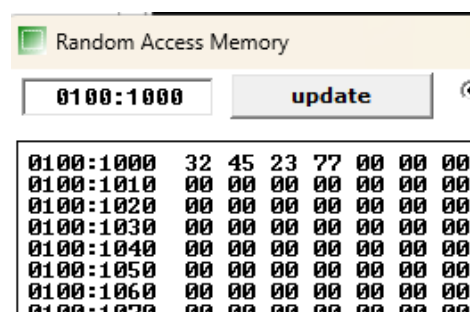
RESULT OF MEMORY

INPUT



Address	Value
0100:1000	32 45 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0100:1010	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0100:1020	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0100:1030	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0100:1040	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0100:1050	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0100:1060	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0100:1070	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

OUTPUT



Address	Value
0100:1000	32 45 23 77 00 00 00 00
0100:1010	00 00 00 00 00 00 00 00
0100:1020	00 00 00 00 00 00 00 00
0100:1030	00 00 00 00 00 00 00 00
0100:1040	00 00 00 00 00 00 00 00
0100:1050	00 00 00 00 00 00 00 00
0100:1060	00 00 00 00 00 00 00 00
0100:1070	00 00 00 00 00 00 00 00

Objective-3 : Find the Gray code of an 8-bit binary number.

```
mov al, [1000h]
mov bl, al
shr al, 01
xor al, bl
mov [1001h], al
hlt
```

RESULT OF MEMORY

Input

Output

Random Access Memory

0100:1000 update

0100:1000	05	00	00	00	00	00	00
0100:1010	00	00	00	00	00	00	00
0100:1020	00	00	00	00	00	00	00
0100:1030	00	00	00	00	00	00	00
0100:1040	00	00	00	00	00	00	00
0100:1050	00	00	00	00	00	00	00
0100:1060	00	00	00	00	00	00	00
0100:1070	00	00	00	00	00	00	00

Random Access Memory

start stop

0100:1000	05	07	00	00	00	00
0100:1010	00	00	00	00	00	00
0100:1020	00	00	00	00	00	00
0100:1030	00	00	00	00	00	00
0100:1040	00	00	00	00	00	00
0100:1050	00	00	00	00	00	00
0100:1060	00	00	00	00	00	00
0100:1070	00	00	00	00	00	00

Objective 4: Find the 2's complement of an 8 -bit number.

```
mov al, [1000h]
not al
add al, 01h
mov [1001h], al
hlt
```

RESULT OF MEMORY

Input

Output

The screenshot shows the "Random Access Memory" window. It features a search bar at the top containing "0100:1000". To the right of the search bar is an "update" button. Below the search bar, a table displays memory contents:

0100:1000	45	BB	00	00	00
0100:1010	00	00	00	00	00
0100:1020	00	00	00	00	00
0100:1030	00	00	00	00	00
0100:1040	00	00	00	00	00
0100:1050	00	00	00	00	00
0100:1060	00	00	00	00	00