LAB REPORT FOR EXP 2

COURSE TITLE : EEE 416

Name : Md Maisoon Rahman
ID : 1606038
Sec : A2

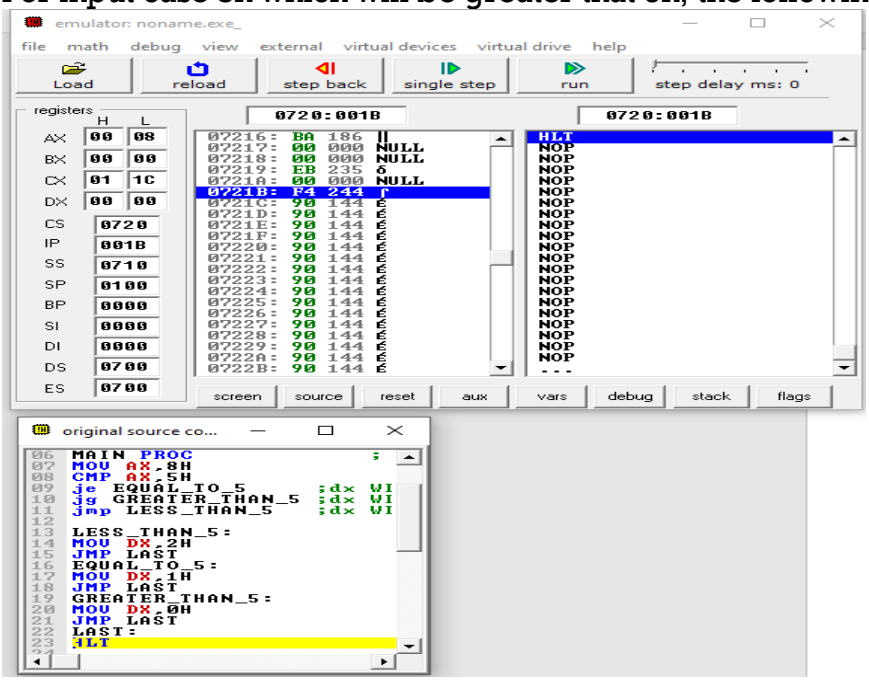
HOMEWORK REPORT 1

CODE :

```
.MODEL SMALL              ;depends on the size of the code
.STACK 100H               ; 100h hexadecimal  stack memory
.DATA

.CODE
MAIN PROC                 ; almost like int main
    MOV AX,3H
    CMP AX,5H
    je EQUAL_TO_5         ;dx WILL BE 1
    jg GREATER_THAN_5     ;dx WILL BE 0
    jmp LESS_THAN_5       ;dx WILL BE 2

LESS_THAN_5:
    MOV DX,2H
    JMP LAST
EQUAL_TO_5:
    MOV DX,1H
    JMP LAST
GREATER_THAN_5:
    MOV DX,0H
    JMP LAST
LAST:
    HLT

    MAIN ENDP
END MAIN ;
```

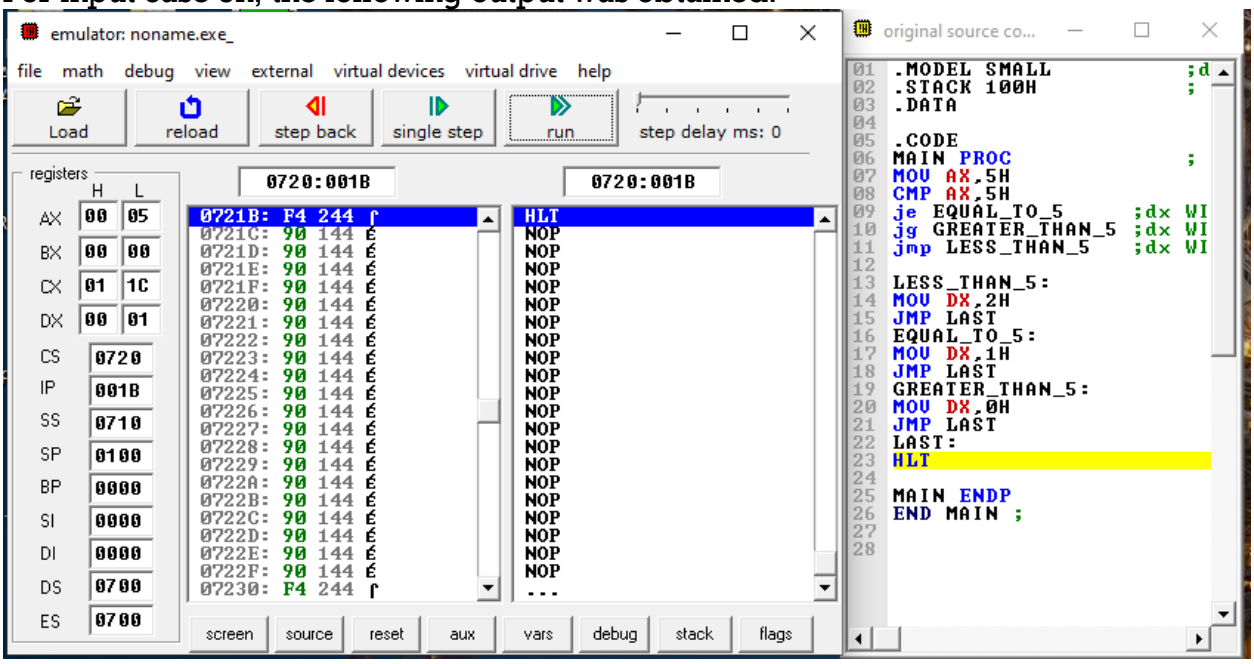
OUTPUT :

For input case 8h which will be greater that 5h, the following result was obtained



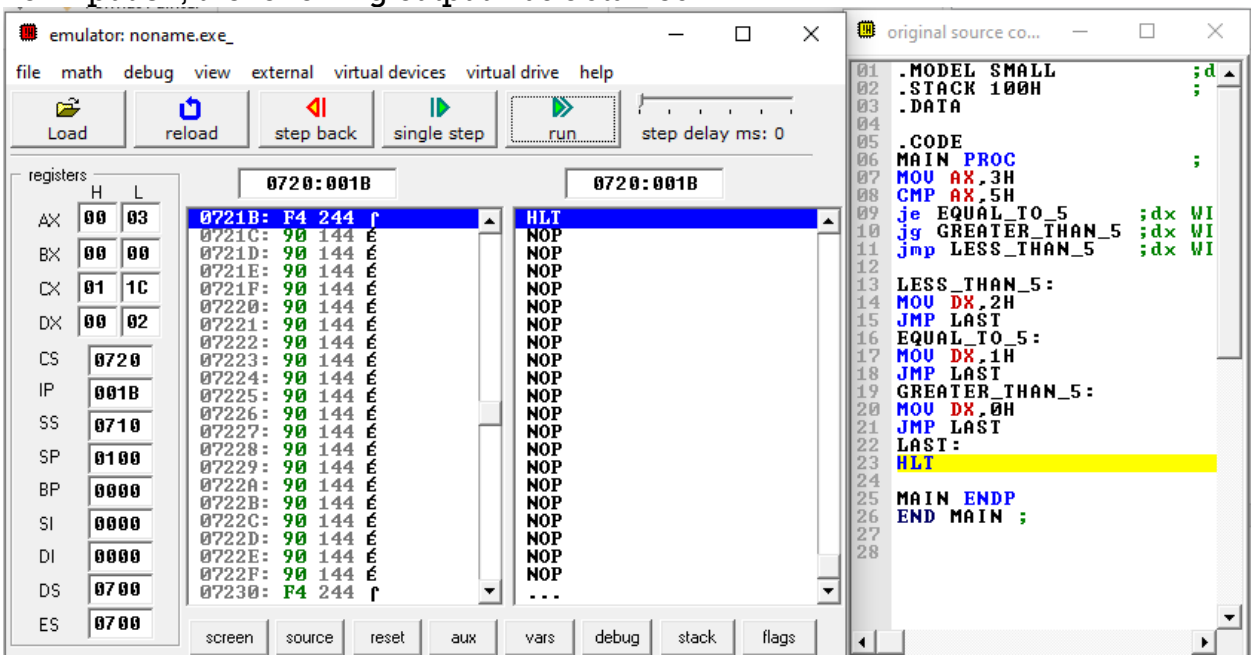
Here, as required, 0 is assigned to DX, as the number is greater than 5h.

For input case 5h, the following output was obtained:



Here, as required, 1 is assigned to DX, as the number is equal to 5h

For input 3h, the following output was obtained:



Here, as required, 2 is assigned to DX, as the number is less than 5h

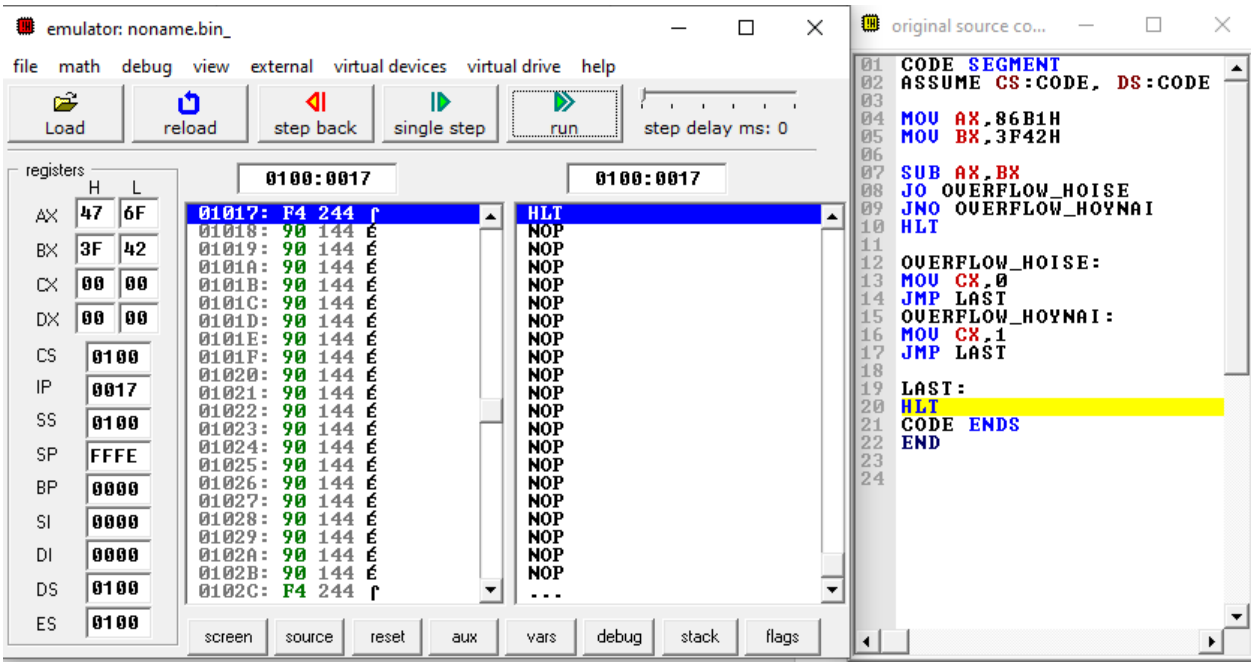
HOMEWORK REPORT 2

CODE:

```
CODE SEGMENT
 ASSUME CS:CODE, DS:CODE

 MOV AX,86B1H
 MOV BX,3F42H

 SUB AX,BX
 JO OVERFLOW_HOISE
 JNO OVERFLOW_HOYNAI
 HLT

OVERFLOW_HOISE:
 MOV CX,0
 JMP LAST
OVERFLOW_HOYNAI:
 MOV CX,1
 JMP LAST

LAST:
 HLT
CODE ENDS
 END
```

OUTPUT:

For 86B1 - 3F42 , there will be overflow.
Therefore, for overflow, register CX should be 0. The desired output was observed in our simulation.

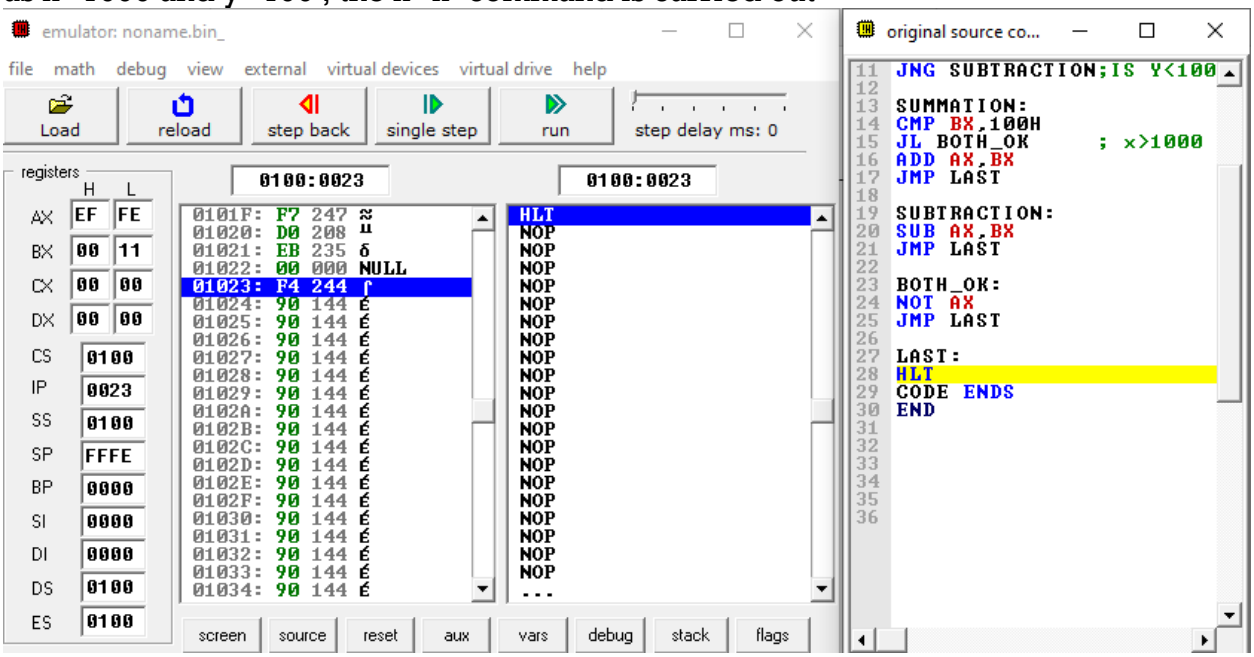


HOMEWORK REPORT 3

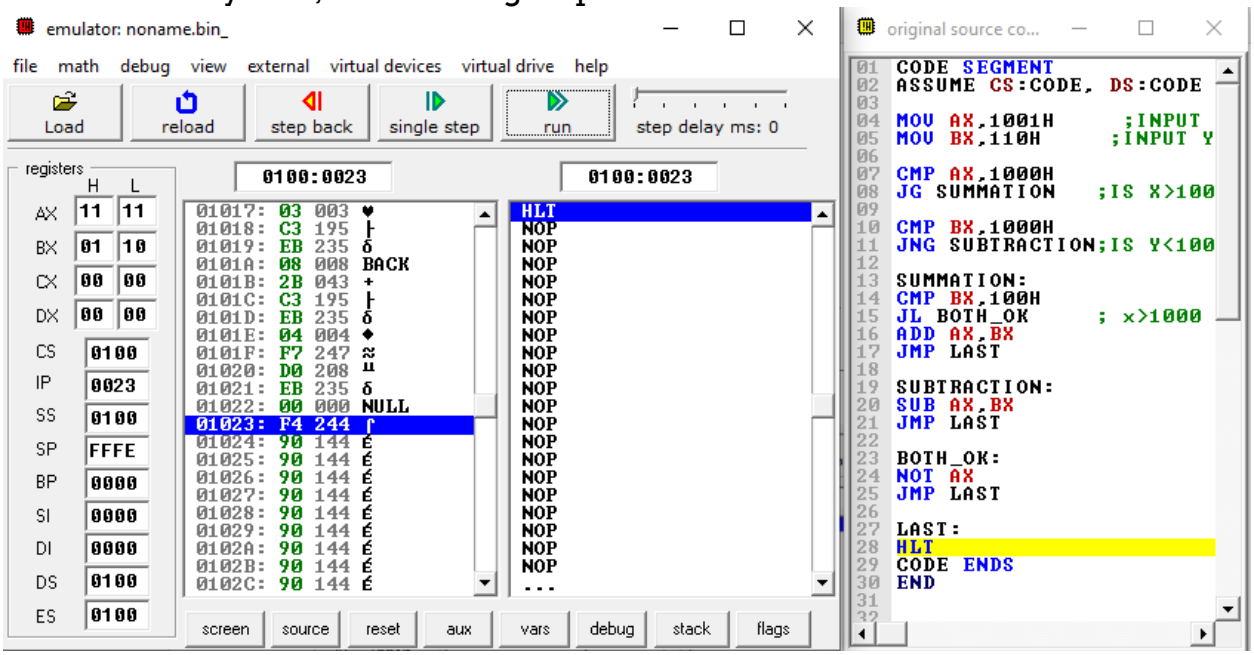
CODE:

```
CODE SEGMENT
 ASSUME CS:CODE, DS:CODE

 MOV AX,10H       ;INPUT X
 MOV BX,11H       ;INPUT Y

 CMP AX,1000H
 JG SUMMATION     ;IS X>1000?

 CMP BX,1000H
 JNG SUBTRACTION;IS Y<1000?

SUMMATION:
 CMP BX,100H
 JL BOTH_OK       ; x>1000 AND Y<100
 ADD AX,BX
 JMP LAST

SUBTRACTION:
 SUB AX,BX
 JMP LAST

BOTH_OK:
 NOT AX
 JMP LAST

LAST:
 HLT
CODE ENDS
 END
```

OUTPUT:
For output x=1001 and y=11;
as x>1000 and y<100 ; the x=x' command is carried out

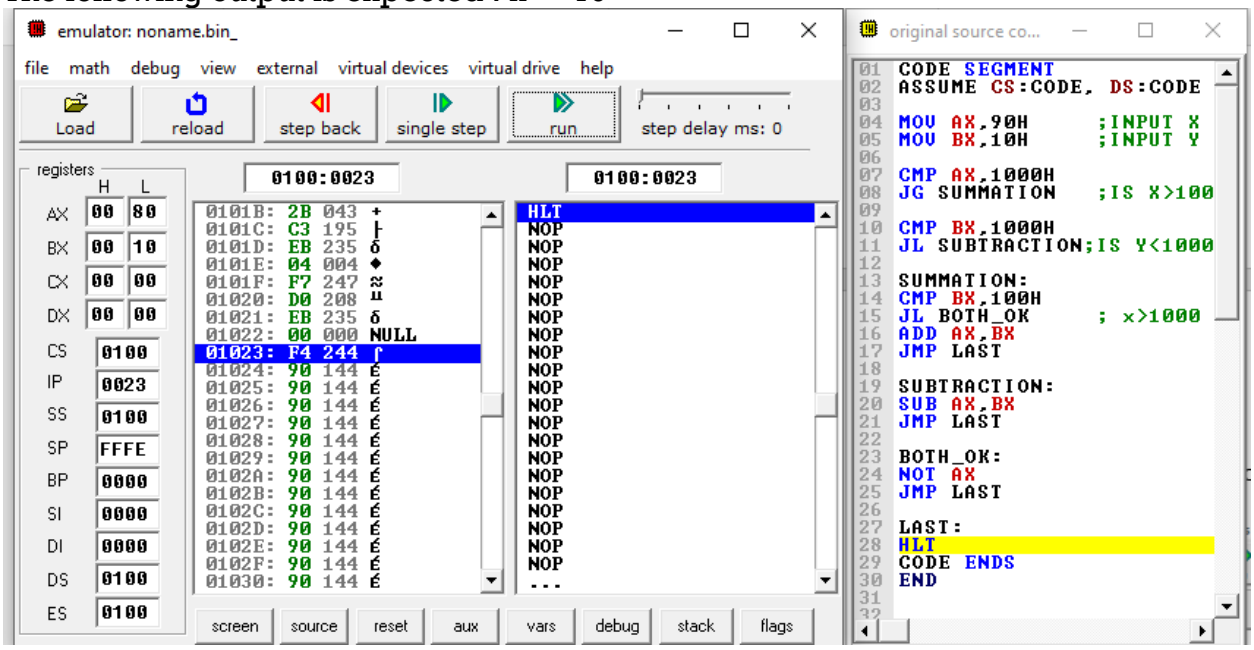


For x=1000 and y=110;
as x>1000 and y>100, the following output was obtained.



As expected, x+y command is carried out.

For x=90 , y=80 ; the following command should have been carried out
x=x-y
The following output is expected : x = 10



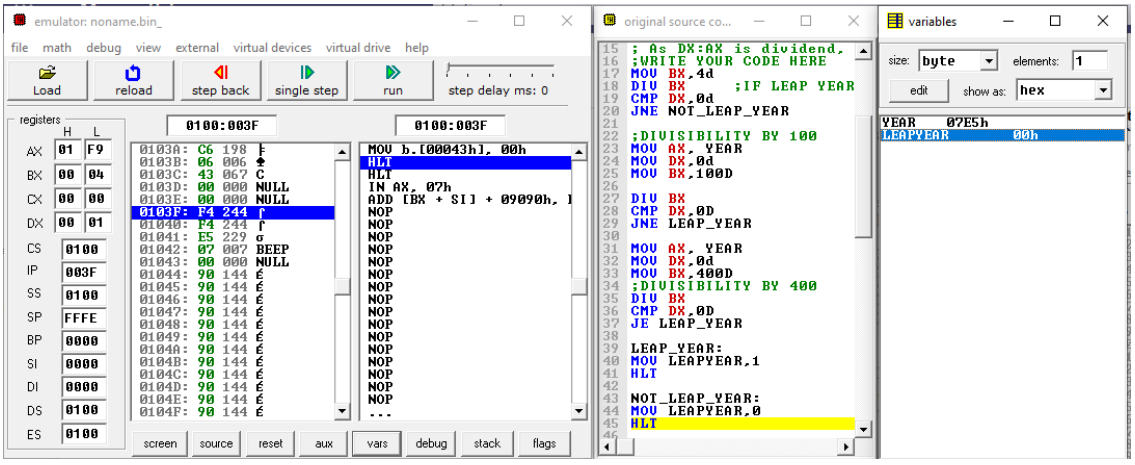
LEAP YEAR

CODE:

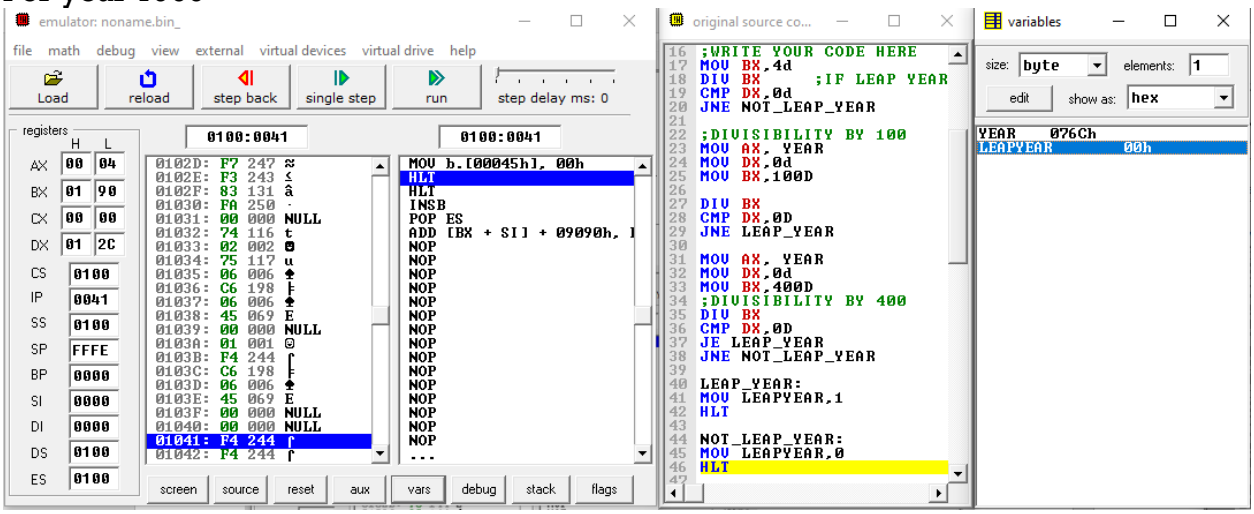
```
CODE SEGMENT

ASSUME CS:CODE, DS:CODE

MOV AX,CS ; these two lines are here to ensure DS=CS. Remember, MOV DS,CS is illegal

MOV DS,AX ; you may ignore these line in EMU. But they are necessary in Hardware


MOV AX, YEAR

MOV DX,0 ; be careful to put 0 in DX before you use DIV in case of 16 bit operand

; As DX:AX is dividend, if there is garbage value in DX, youÆll get ; specious result
;WRITE YOUR CODE HERE
    MOV BX,4d
    DIV BX      ;IF LEAP YEAR, REMAINDER DX =0
    CMP DX,0d
    JNE NOT_LEAP_YEAR

    ;DIVISIBILITY BY 100
    MOV AX, YEAR
    MOV DX,0d
    MOV BX,100D

    DIV BX
    CMP DX,0D
    JNE LEAP_YEAR

    MOV AX, YEAR
    MOV DX,0d
    MOV BX,400D
    ;DIVISIBILITY BY 400
    DIV BX
    CMP DX,0D
    JE LEAP_YEAR
    JNE NOT_LEAP_YEAR

LEAP_YEAR:
    MOV LEAPYEAR,1
    HLT

NOT_LEAP_YEAR:
    MOV LEAPYEAR,0
    HLT


HLT


YEAR DW 1900D

LEAPYEAR DB ?

CODE ENDS

END
```

OUTPUT:

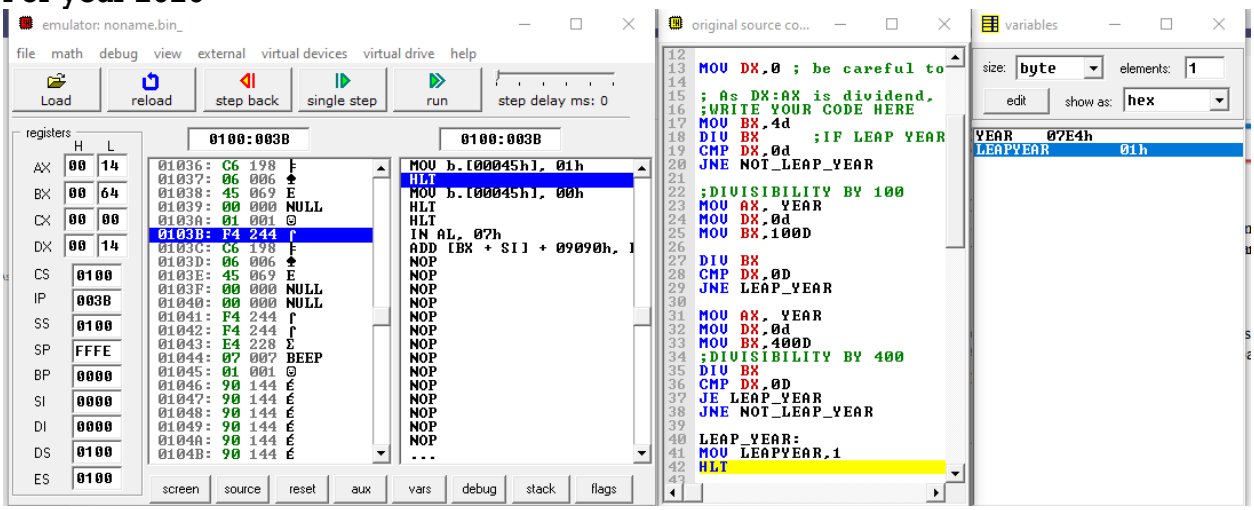
For Year=2021



For year 1900



For year 2020



For year 2000