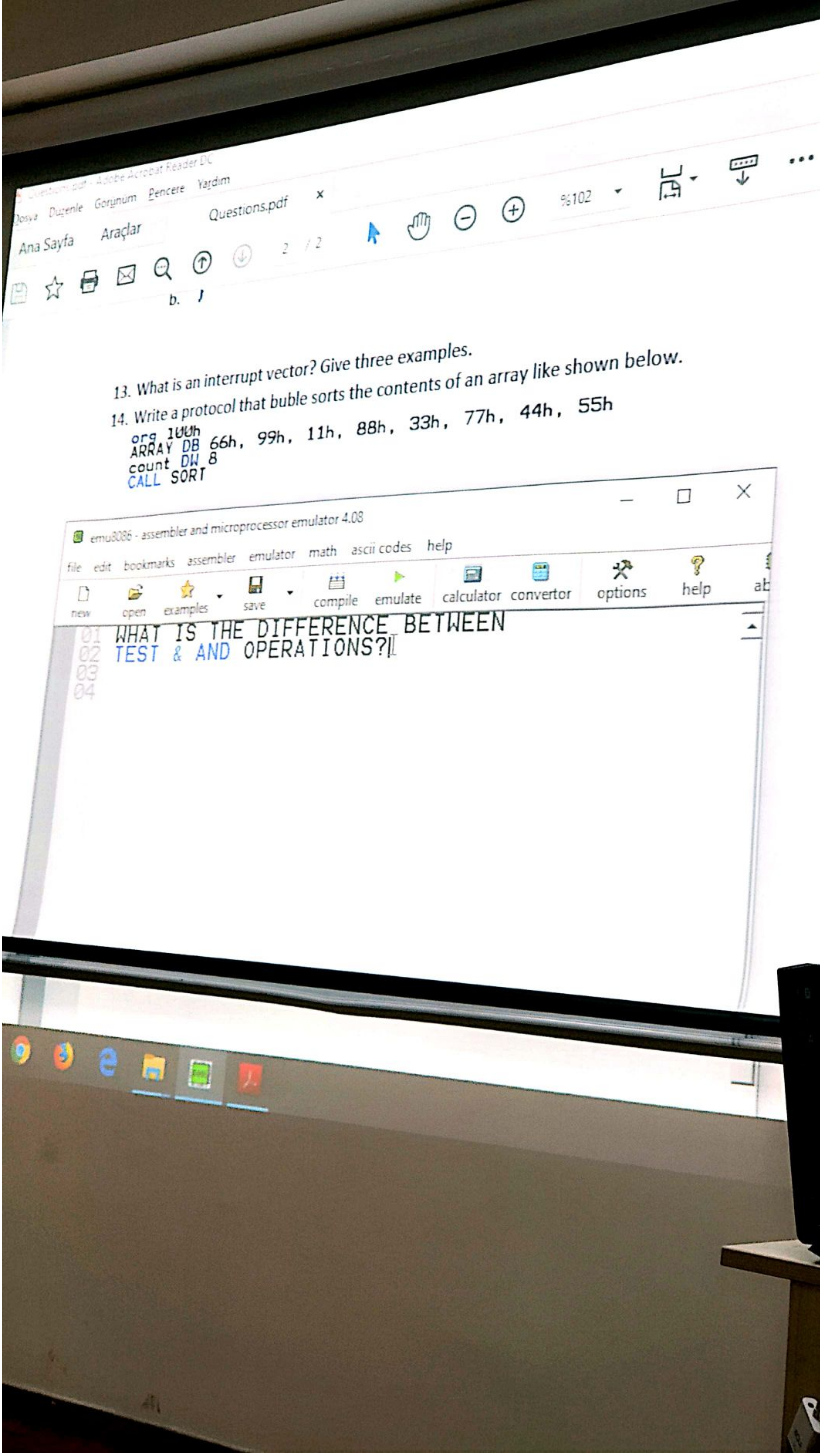

Questions.pdf - Adobe Acrobat Reader DC
Dosya Düzenle Görünüm Pencere Yardım
Ana Sayfa Araçlar Questions.pdf
2 / 2
%102

b. J

13. What is an interrupt vector? Give three examples.

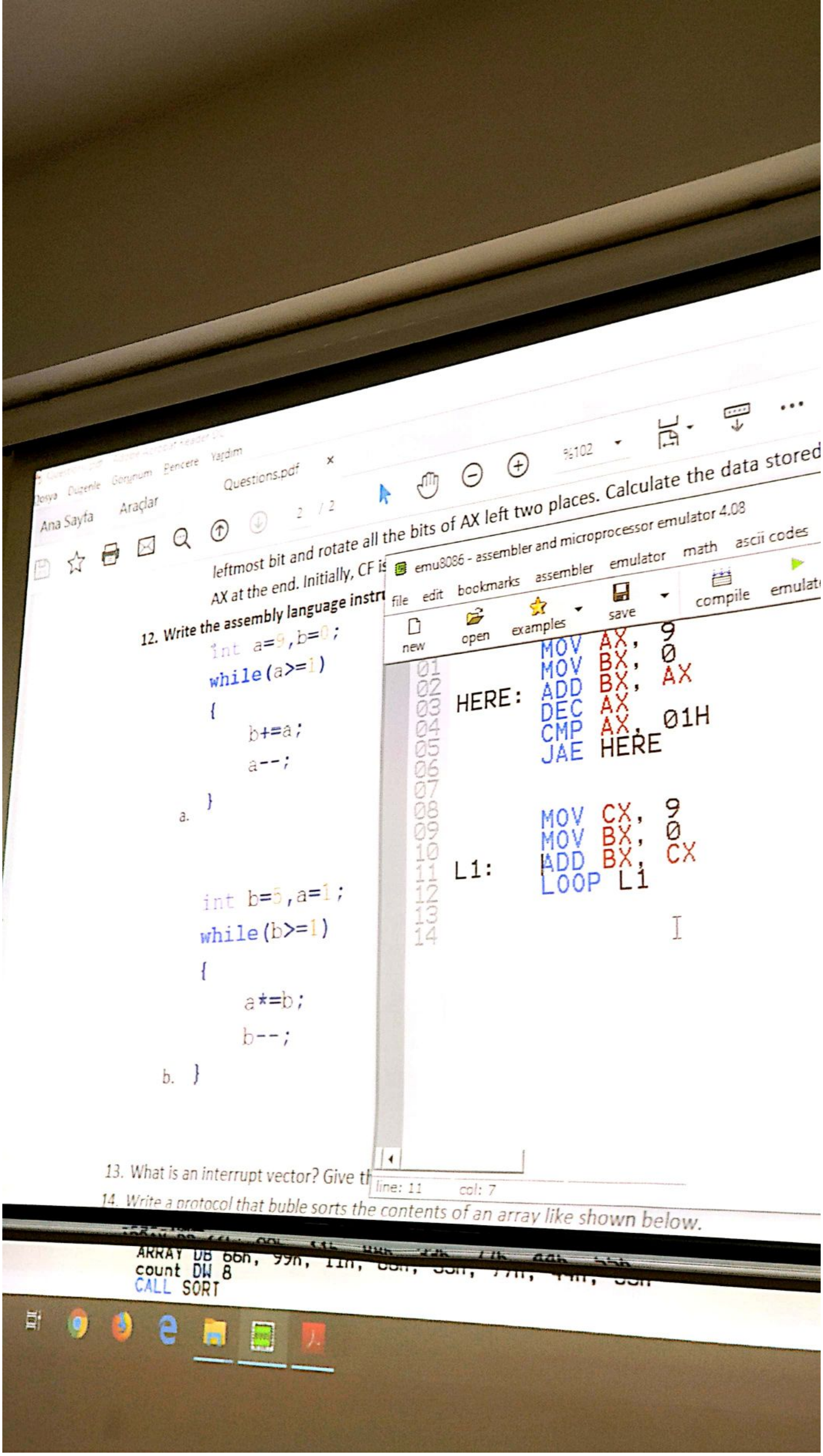
14. Write a protocol that buble sorts the contents of an array like shown below.

```
org 100h
ARRAY DB 66h, 99h, 11h, 88h, 33h, 77h, 44h, 55h
count DW 8
CALL SORT
```

emu8086 - assembler and microprocessor emulator 4.08
file edit bookmarks assembler emulator math ascii codes help
new open examples save compile emulate calculator convertor options help

```
WHAT IS THE DIFFERENCE BETWEEN
TEST & AND OPERATIONS?
```

Scanned by CamScanner


Dosya Düzenle Görünüm Pencere Yardım
Ana Sayfa
Araçlar
Questions.pdf
2 / 2
%102
emu8086 - assembler and microprocessor emulator 4.08
file edit bookmarks assembler emulator math ascii codes
new open examples save compile emulate
line: 11 col: 7

leftmost bit and rotate all the bits of AX left two places. Calculate the data stored

AX at the end. Initially, CF is

12. Write the assembly language instru

a.
```
int a=9,b=0;
while(a>=1)
{
    b+=a;
    a--;
}
```

b.
```
int b=5,a=1;
while(b>=1)
{
    a*=b;
    b--;
}
```

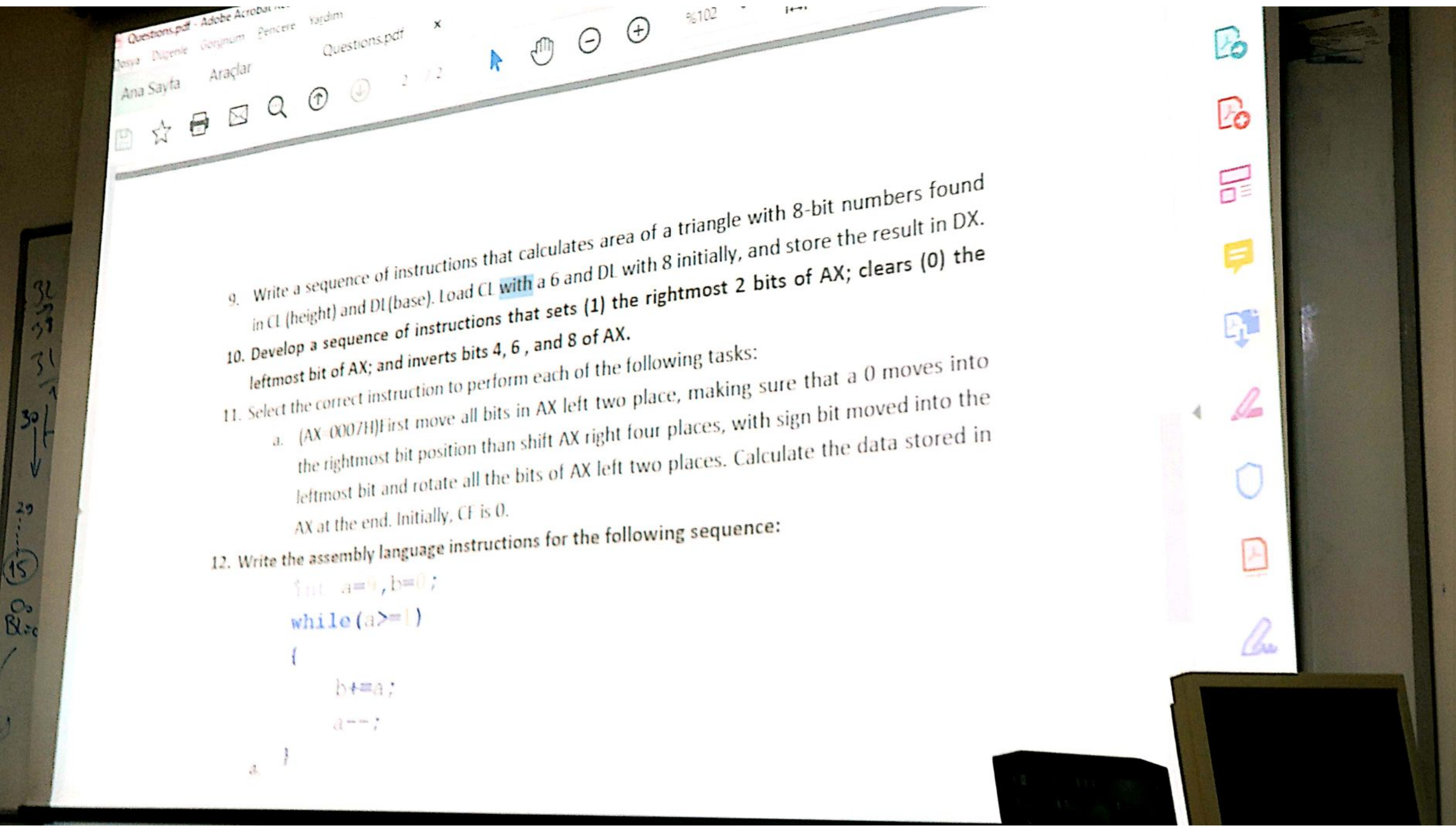
```
        MOV AX, 9
        MOV BX, 0
HERE:   ADD BX, AX
        DEC AX
        CMP AX, 01H
        JAE HERE


        MOV CX, 9
        MOV BX, 0
L1:     ADD BX, CX
        LOOP L1
```

13. What is an interrupt vector? Give th

14. Write a protocol that buble sorts the contents of an array like shown below.

```
ARRAY DB 66h, 99h, 11h, 88h, 33h, 77h, 44h, 55h
count DW 8
CALL SORT
```


Questions.pdf - Adobe Acrobat
Dosya Düzenle Görünüm Pencere Yardım
Ana Sayfa
Araçlar
Questions.pdf
2 / 2
%102

9. Write a sequence of instructions that calculates area of a triangle with 8-bit numbers found in CL (height) and DL (base). Load CL with a 6 and DL with 8 initially, and store the result in DX.

10. Develop a sequence of instructions that sets (1) the rightmost 2 bits of AX; clears (0) the leftmost bit of AX; and inverts bits 4, 6 , and 8 of AX.

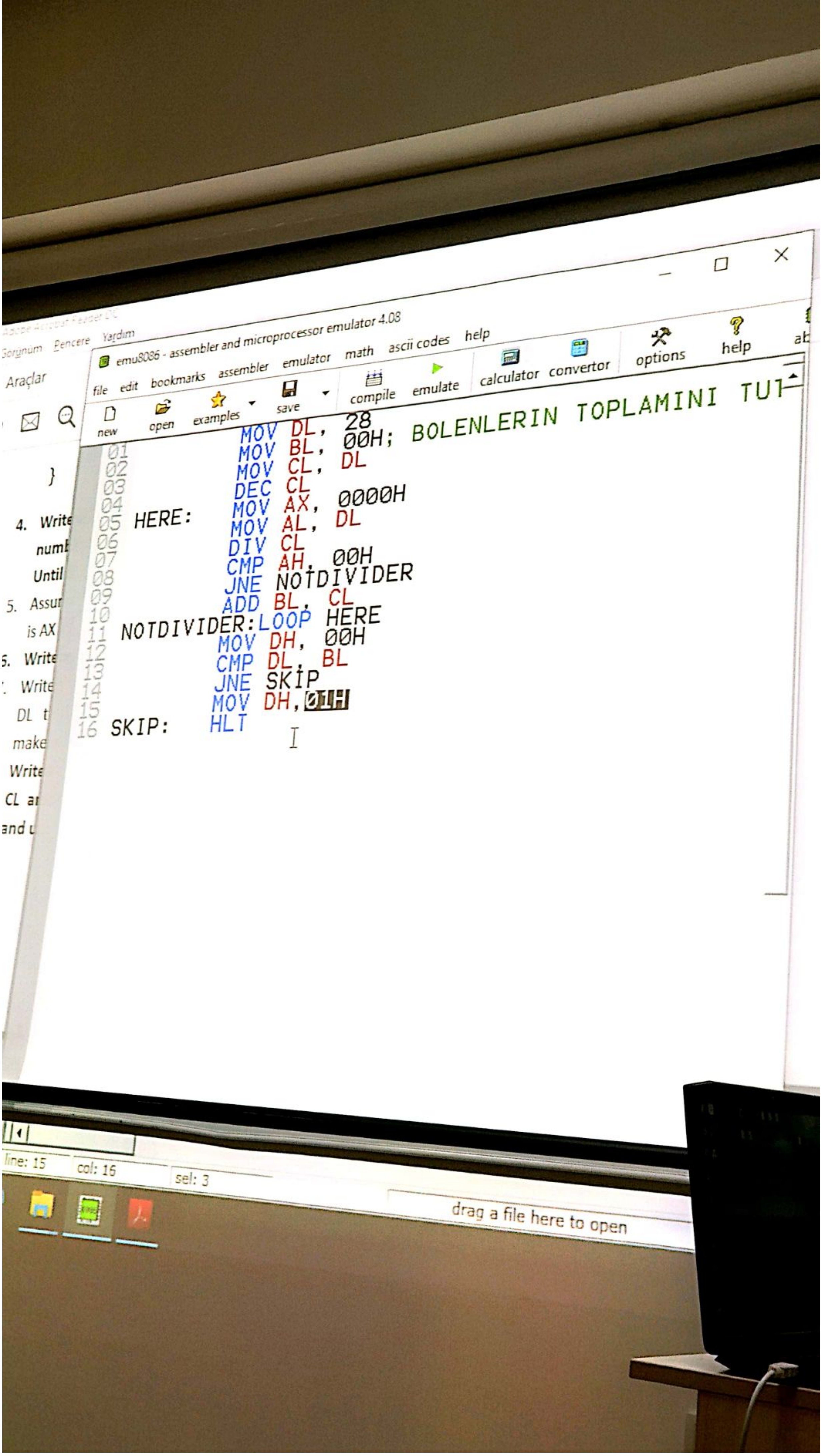
11. Select the correct instruction to perform each of the following tasks:

   a. (AX=0007H)First move all bits in AX left two place, making sure that a 0 moves into the rightmost bit position than shift AX right four places, with sign bit moved into the leftmost bit and rotate all the bits of AX left two places. Calculate the data stored in AX at the end. Initially, CF is 0.

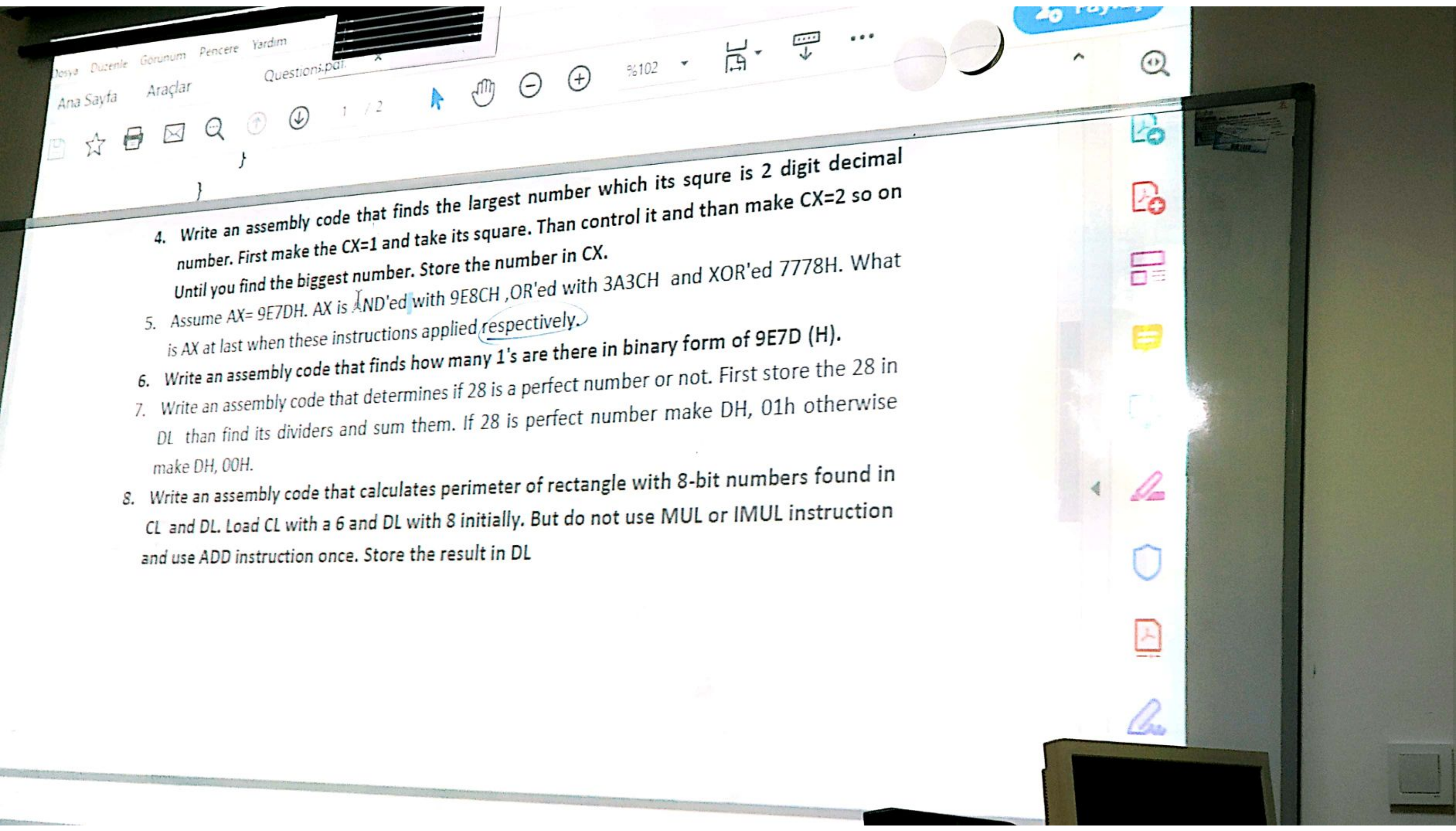
12. Write the assembly language instructions for the following sequence:

```
int a=[illegible],b=0;
while(a>=1)
{
    b+=a;
    a--;
}
```

a.



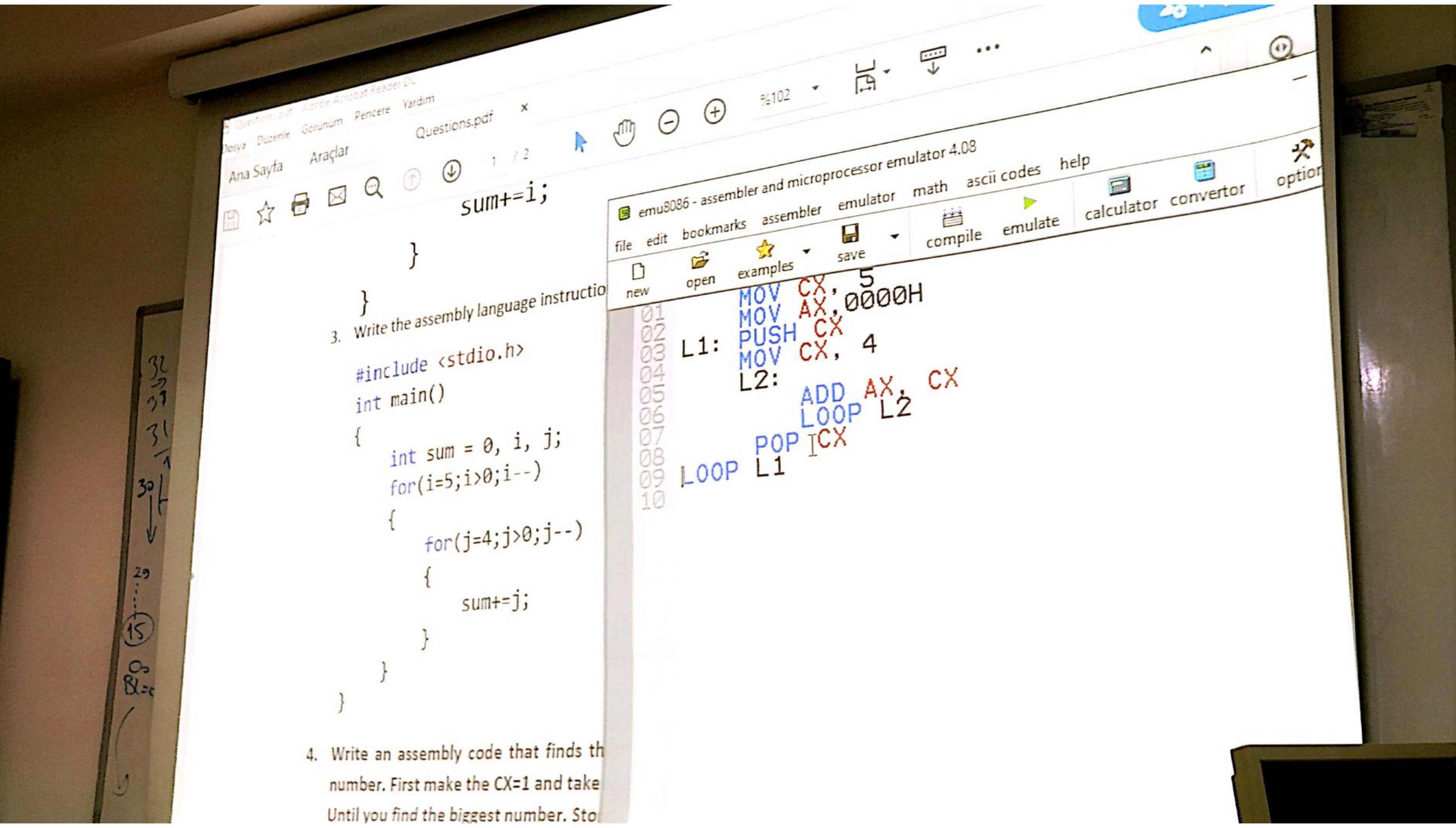
```
          MOV DL, 28
          MOV BL, 00H; BOLENLERIN TOPLAMINI TUT
          MOV CL, DL
          DEC CL
HERE:     MOV AX, 0000H
          MOV AL, DL
          DIV CL
          CMP AH, 00H
          JNE NOTDIVIDER
          ADD BL, CL
NOTDIVIDER:LOOP HERE
          MOV DH, 00H
          CMP DL, BL
          JNE SKIP
          MOV DH,01H
SKIP:     HLT
```



}

}

4. Write an assembly code that finds the largest number which its squre is 2 digit decimal number. First make the CX=1 and take its square. Than control it and than make CX=2 so on Until you find the biggest number. Store the number in CX.
5. Assume AX= 9E7DH. AX is AND'ed with 9E8CH ,OR'ed with 3A3CH and XOR'ed 7778H. What is AX at last when these instructions applied respectively.
6. Write an assembly code that finds how many 1's are there in binary form of 9E7D (H).
7. Write an assembly code that determines if 28 is a perfect number or not. First store the 28 in DL than find its dividers and sum them. If 28 is perfect number make DH, 01h otherwise make DH, 00H.
8. Write an assembly code that calculates perimeter of rectangle with 8-bit numbers found in CL and DL. Load CL with a 6 and DL with 8 initially. But do not use MUL or IMUL instruction and use ADD instruction once. Store the result in DL



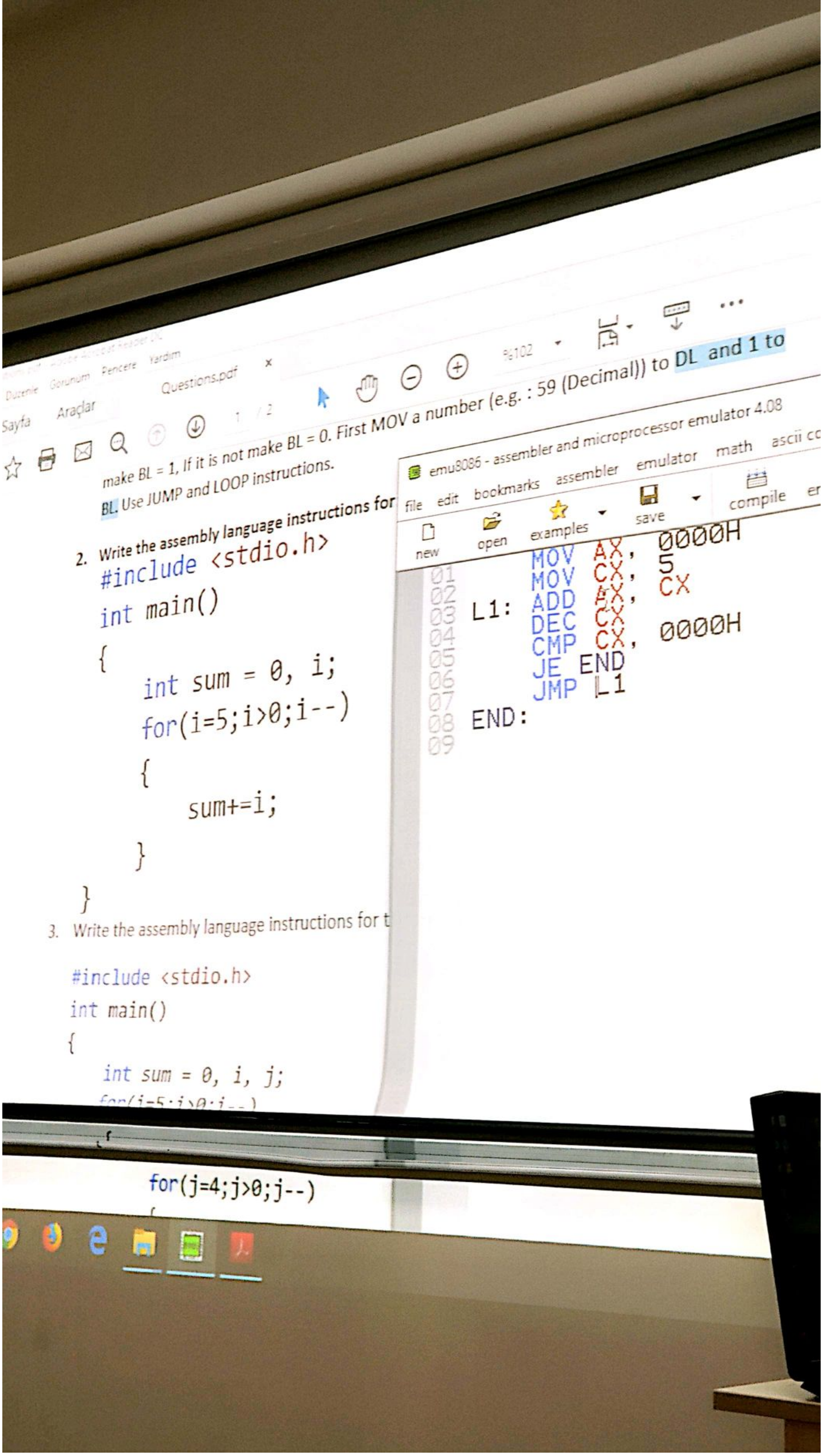
```
        sum+=i;
    }
}
```

3. Write the assembly language instructio

```
#include <stdio.h>
int main()
{
    int sum = 0, i, j;
    for(i=5;i>0;i--)
    {
        for(j=4;j>0;j--)
        {
            sum+=j;
        }
    }
}
```

4. Write an assembly code that finds th
number. First make the CX=1 and take
Until you find the biggest number. Sto

```
    MOV CX, 5
    MOV AX,0000H
L1: PUSH CX
    MOV CX, 4
    L2:
        ADD AX, CX
        LOOP L2
      POP CX
LOOP L1
```



make BL = 1, If it is not make BL = 0. First MOV a number (e.g. : 59 (Decimal)) to DL and 1 to BL. Use JUMP and LOOP instructions.

2. Write the assembly language instructions for

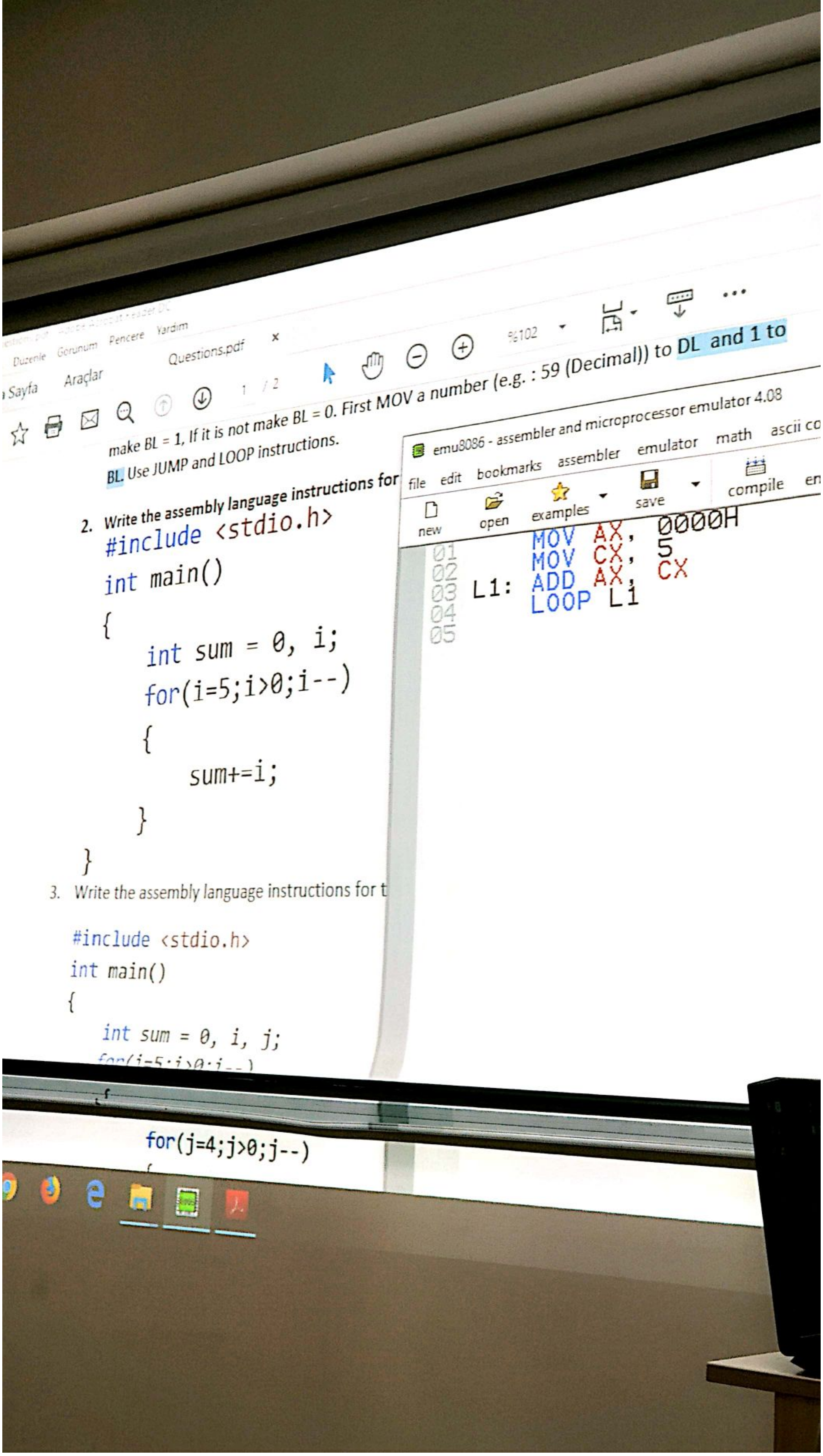
```
#include <stdio.h>
int main()
{
    int sum = 0, i;
    for(i=5;i>0;i--)
    {
        sum+=i;
    }
}
```

```
        MOV AX, 0000H
        MOV CX, 5
L1:     ADD AX, CX
        DEC CX
        CMP CX, 0000H
        JE END
        JMP L1
END:
```

3. Write the assembly language instructions for t

```
#include <stdio.h>
int main()
{
    int sum = 0, i, j;
    for(i=5;i>0;i--)

        for(j=4;j>0;j--)
```


Düzenle Görünüm Pencere Yardım
Sayfa
Araçlar
Questions.pdf
%102
emu8086 - assembler and microprocessor emulator 4.08
file edit bookmarks assembler emulator math ascii co
new open examples save compile

make BL = 1, If it is not make BL = 0. First MOV a number (e.g. : 59 (Decimal)) to DL and 1 to BL. Use JUMP and LOOP instructions.

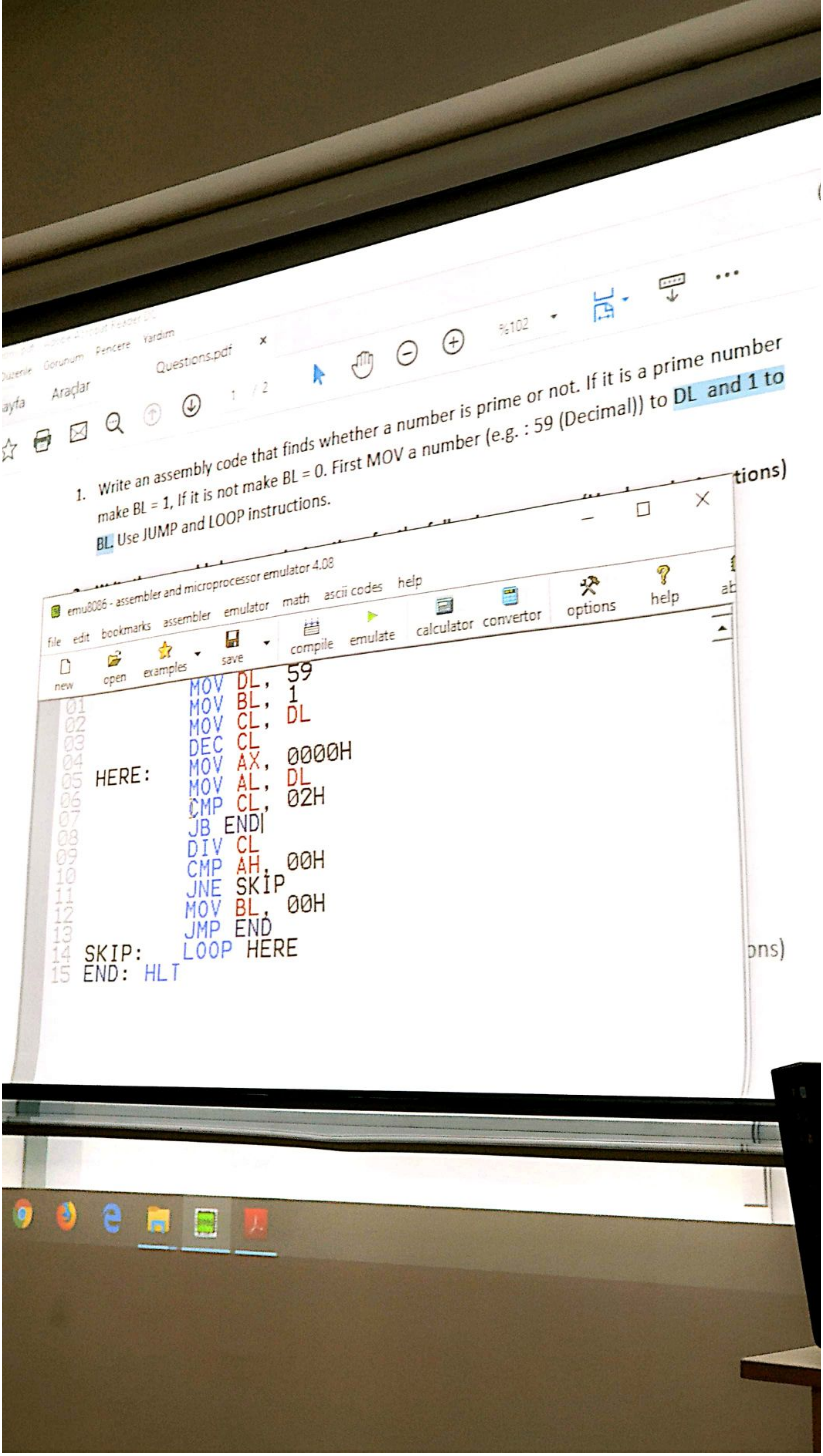
2. Write the assembly language instructions for

```
#include <stdio.h>
int main()
{
    int sum = 0, i;
    for(i=5;i>0;i--)
    {
        sum+=i;
    }
}
```

3. Write the assembly language instructions for t

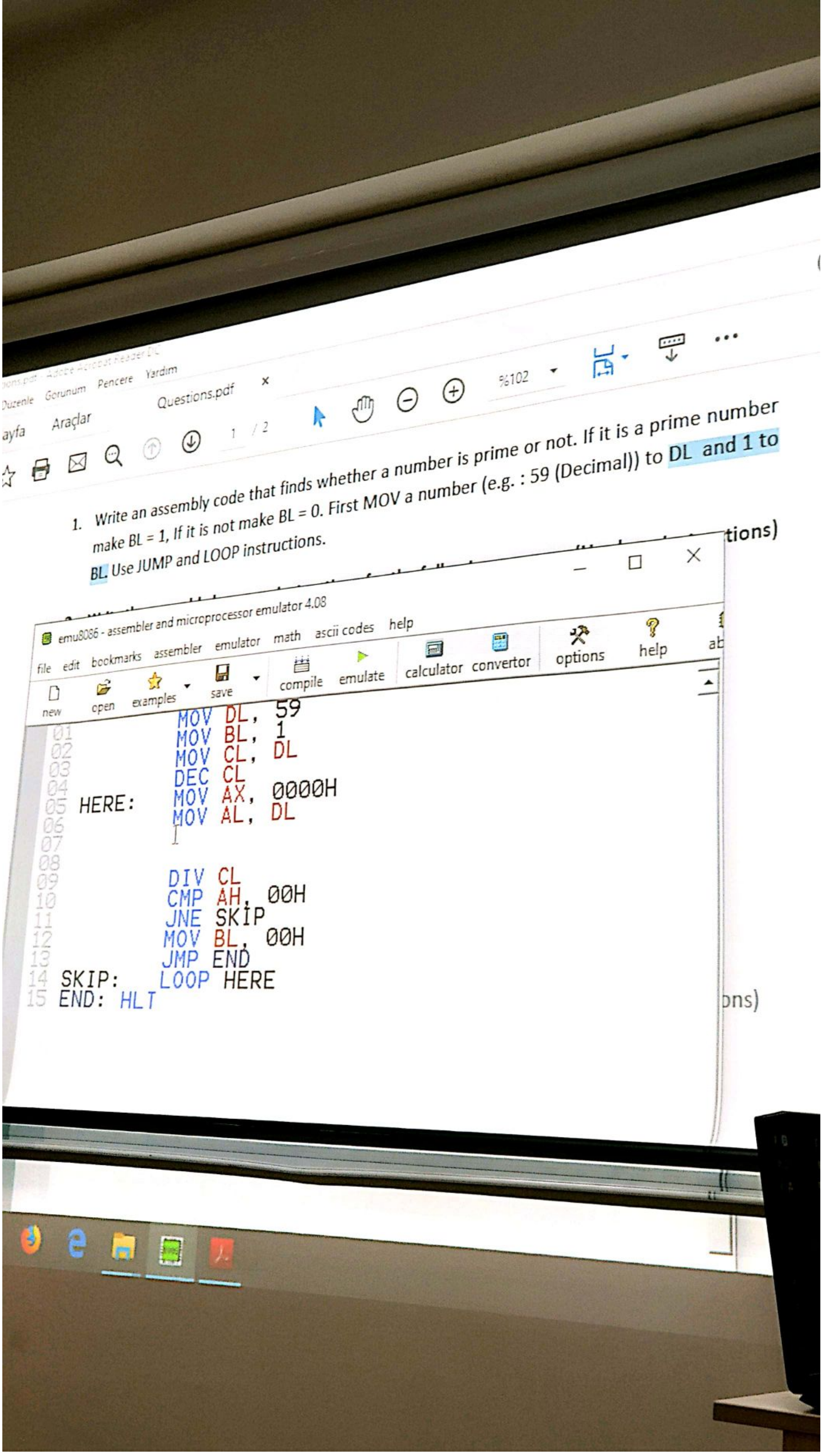
```
#include <stdio.h>
int main()
{
    int sum = 0, i, j;
    for(i=5;i>0;i--)
    {
        for(j=4;j>0;j--)
        {
```

```
    MOV AX, 0000H
    MOV CX, 5
L1: ADD AX, CX
    LOOP L1
```



1. Write an assembly code that finds whether a number is prime or not. If it is a prime number make BL = 1, If it is not make BL = 0. First MOV a number (e.g. : 59 (Decimal)) to DL and 1 to BL. Use JUMP and LOOP instructions.

```
               MOV DL, 59
               MOV BL, 1
               MOV CL, DL
               DEC CL
               MOV AX, 0000H
HERE:          MOV AL, DL
               CMP CL, 02H
               JB END
               DIV CL
               CMP AH, 00H
               JNE SKIP
               MOV BL, 00H
               JMP END
SKIP:          LOOP HERE
END: HLT
```


Questions.pdf
Araçlar
1 / 2
%102
emu8086 - assembler and microprocessor emulator 4.08
file edit bookmarks assembler emulator math ascii codes help
new open examples save compile emulate calculator convertor options help

1. Write an assembly code that finds whether a number is prime or not. If it is a prime number make BL = 1, If it is not make BL = 0. First MOV a number (e.g. : 59 (Decimal)) to DL and 1 to BL. Use JUMP and LOOP instructions.

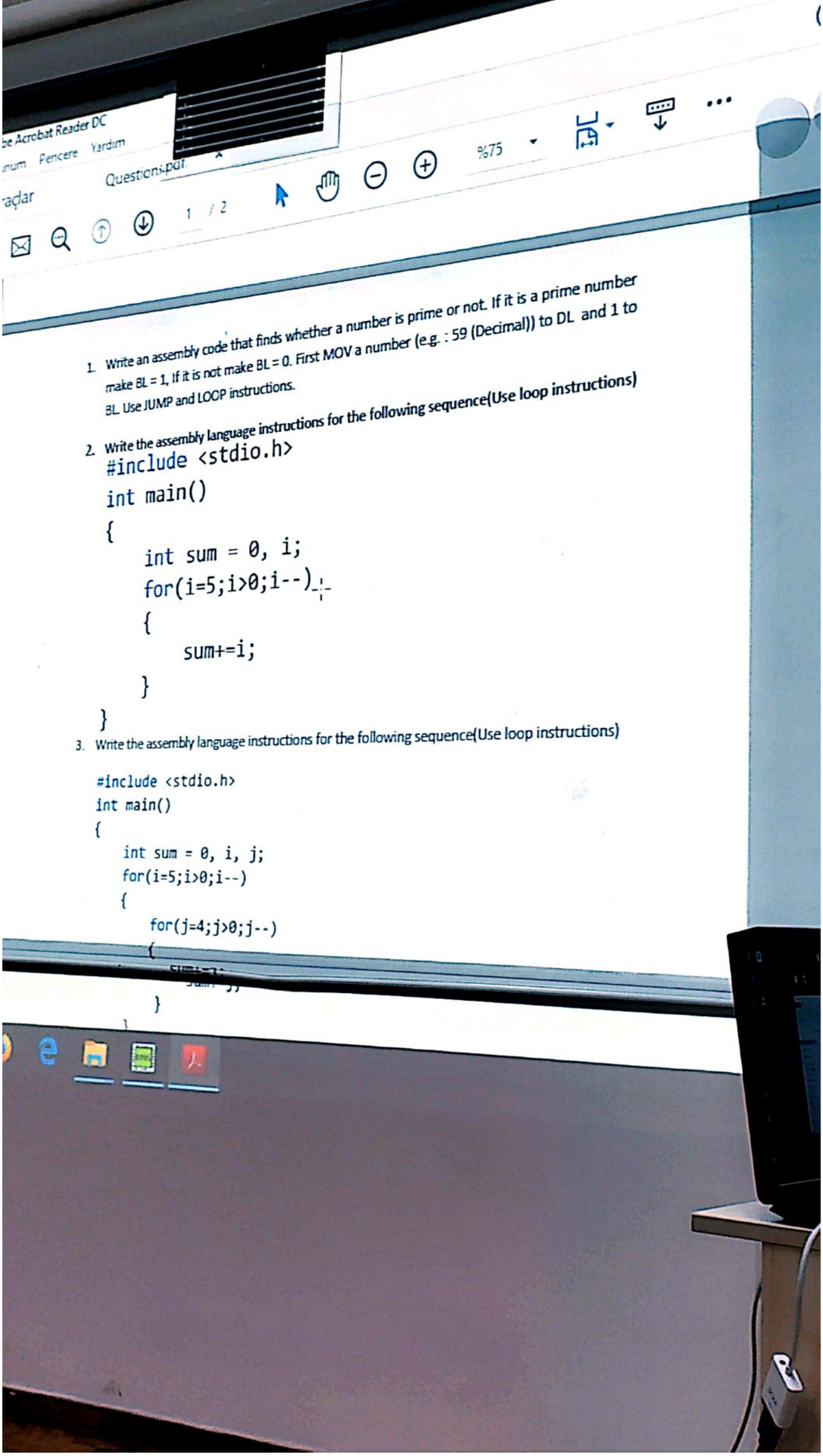
```
          MOV DL, 59
          MOV BL, 1
          MOV CL, DL
          DEC CL
HERE:     MOV AX, 0000H
          MOV AL, DL



          DIV CL
          CMP AH, 00H
          JNE SKIP
          MOV BL, 00H
          JMP END
SKIP:     LOOP HERE
END: HLT
```

tions)

ons)



1. Write an assembly code that finds whether a number is prime or not. If it is a prime number make BL = 1, If it is not make BL = 0. First MOV a number (e.g. : 59 (Decimal)) to DL and 1 to BL. Use JUMP and LOOP instructions.

2. Write the assembly language instructions for the following sequence(Use loop instructions)

```
#include <stdio.h>
int main()
{
    int sum = 0, i;
    for(i=5;i>0;i--)
    {
        sum+=i;
    }
}
```

3. Write the assembly language instructions for the following sequence(Use loop instructions)

```
#include <stdio.h>
int main()
{
    int sum = 0, i, j;
    for(i=5;i>0;i--)
    {
        for(j=4;j>0;j--)
        {
            sum+=j;
        }
```