Daffodil International University

Lab Report

Course Name: Microprocessor, Embedded Systems & IoT lab
Course Code: CSE 232

Submitted By:

Mahadi Hasan Munna

ID: 203-15-3881
Section: PC – A
Department of CSE

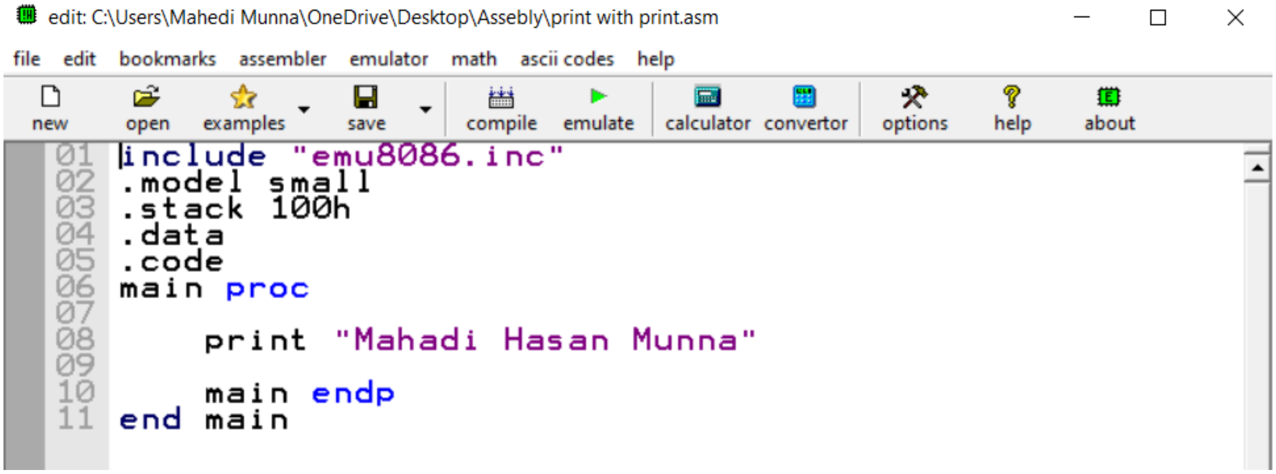
Submitted To:

Sonia Nasrin

Lecturer
Department of CSE
Daffodil International University

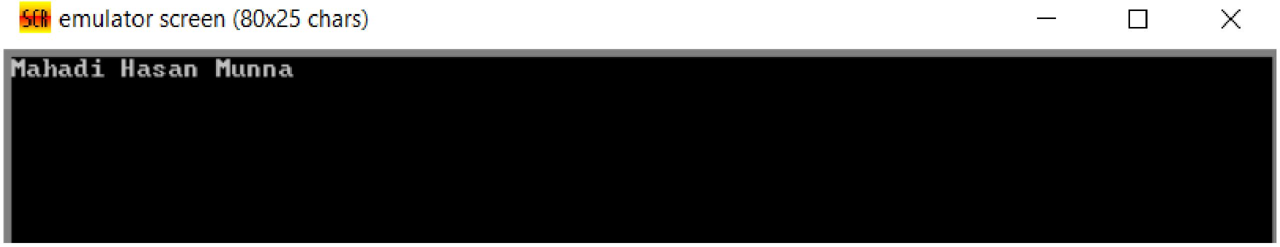
Lab 1

1. Printing my name with print function.



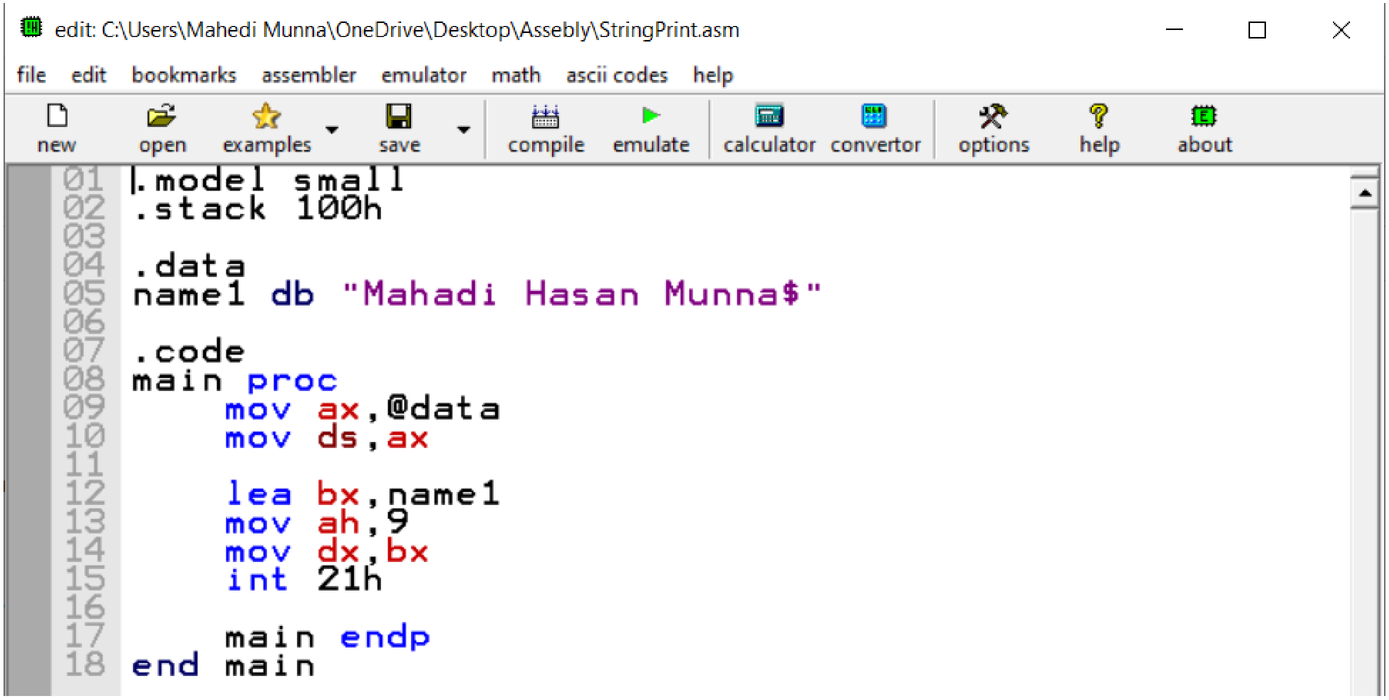
```
include "emu8086.inc"
.model small
.stack 100h
.data
.code
main proc

      print "Mahadi Hasan Munna"

      main endp
end main
```

Output:



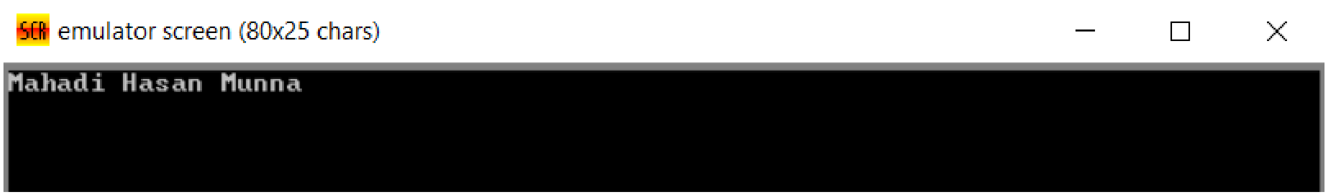
```
Mahadi Hasan Munna
```

2. Printing my name with assembly



```
.model small
.stack 100h

.data
name1 db "Mahadi Hasan Munna$"

.code
main proc
      mov ax,@data
      mov ds,ax

      lea bx,name1
      mov ah,9
      mov dx,bx
      int 21h

      main endp
end main
```

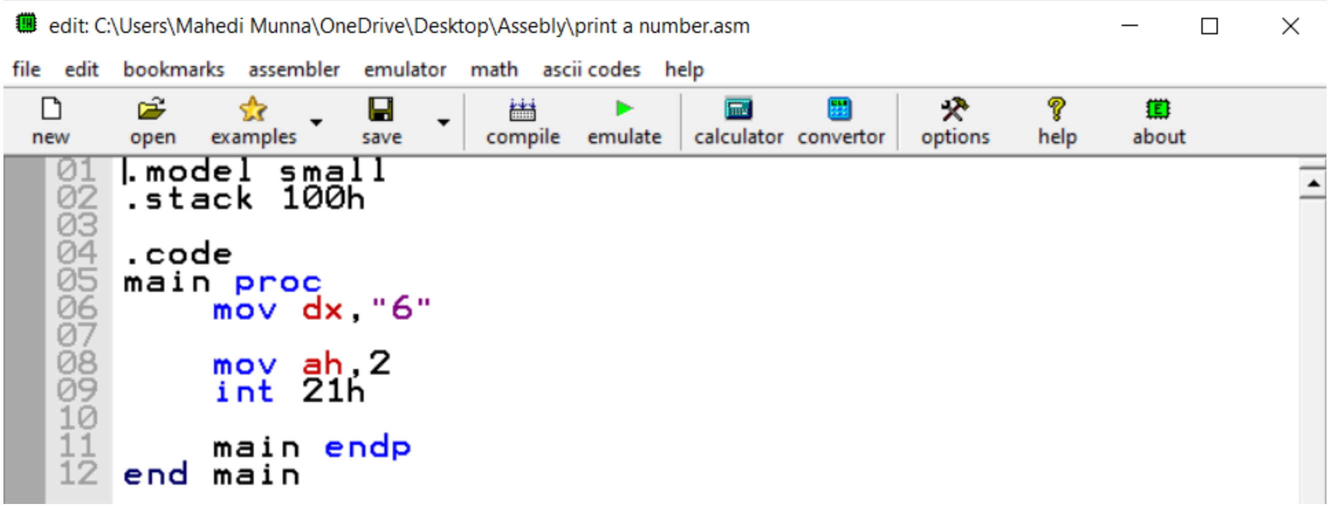
Output:



```
Mahadi Hasan Munna
```

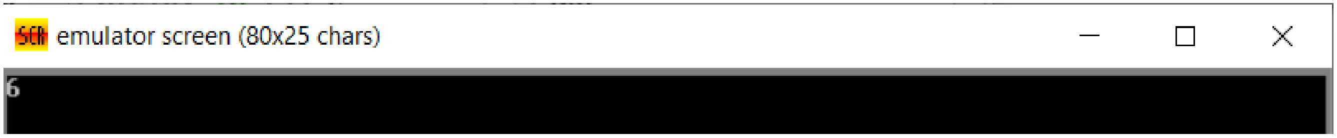
Lab 2

1. Print a number



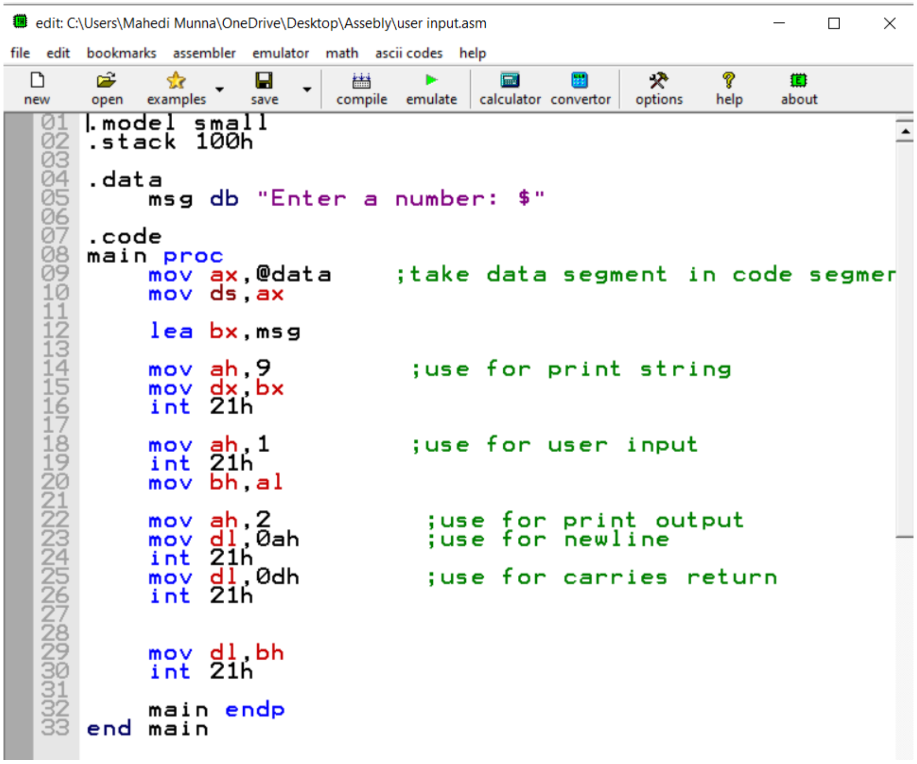
```
.model small
.stack 100h

.code
main proc
    mov dx,"6"

    mov ah,2
    int 21h

    main endp
end main
```

Output:



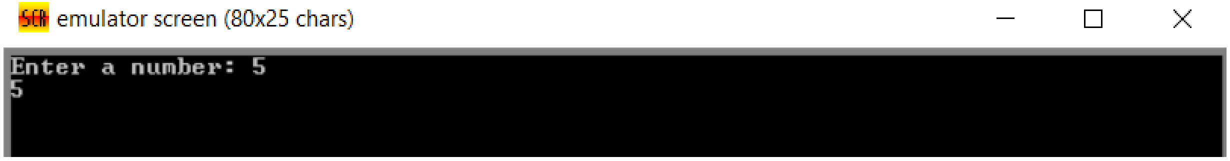
```
6
```

2. Take a user input and print it.



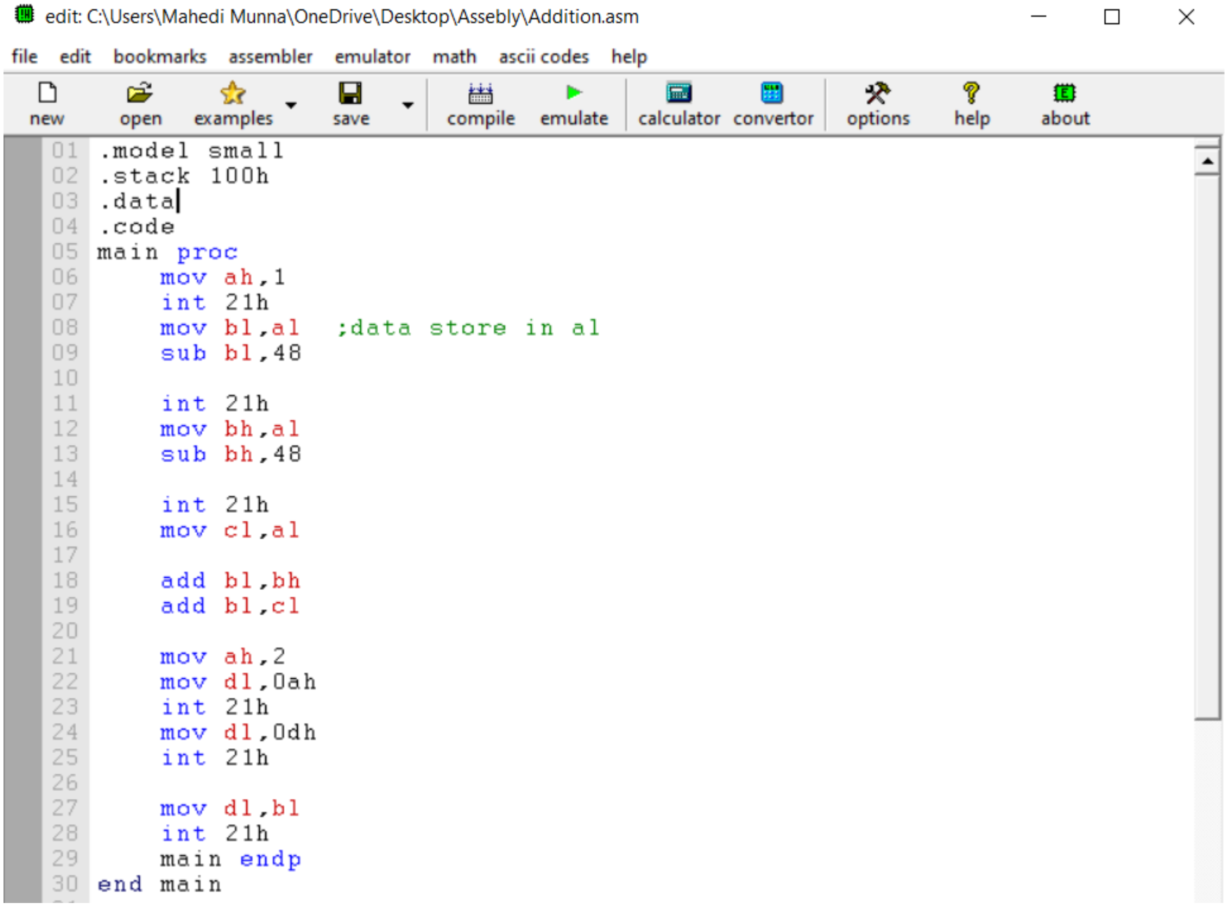
```
.model small
.stack 100h

.data
    msg db "Enter a number: $"

.code
main proc
    mov ax,@data    ;take data segment in code segmer
    mov ds,ax

    lea bx,msg

    mov ah,9         ;use for print string
    mov dx,bx
    int 21h

    mov ah,1         ;use for user input
    int 21h
    mov bh,al

    mov ah,2          ;use for print output
    mov dl,0ah        ;use for newline
    int 21h
    mov dl,0dh        ;use for carries return
    int 21h


    mov dl,bh
    int 21h

    main endp
end main
```

Output:



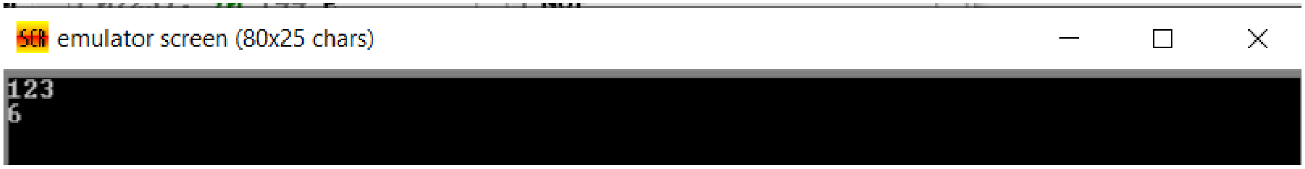
```
Enter a number: 5
5
```

3. Add 3 Numbers



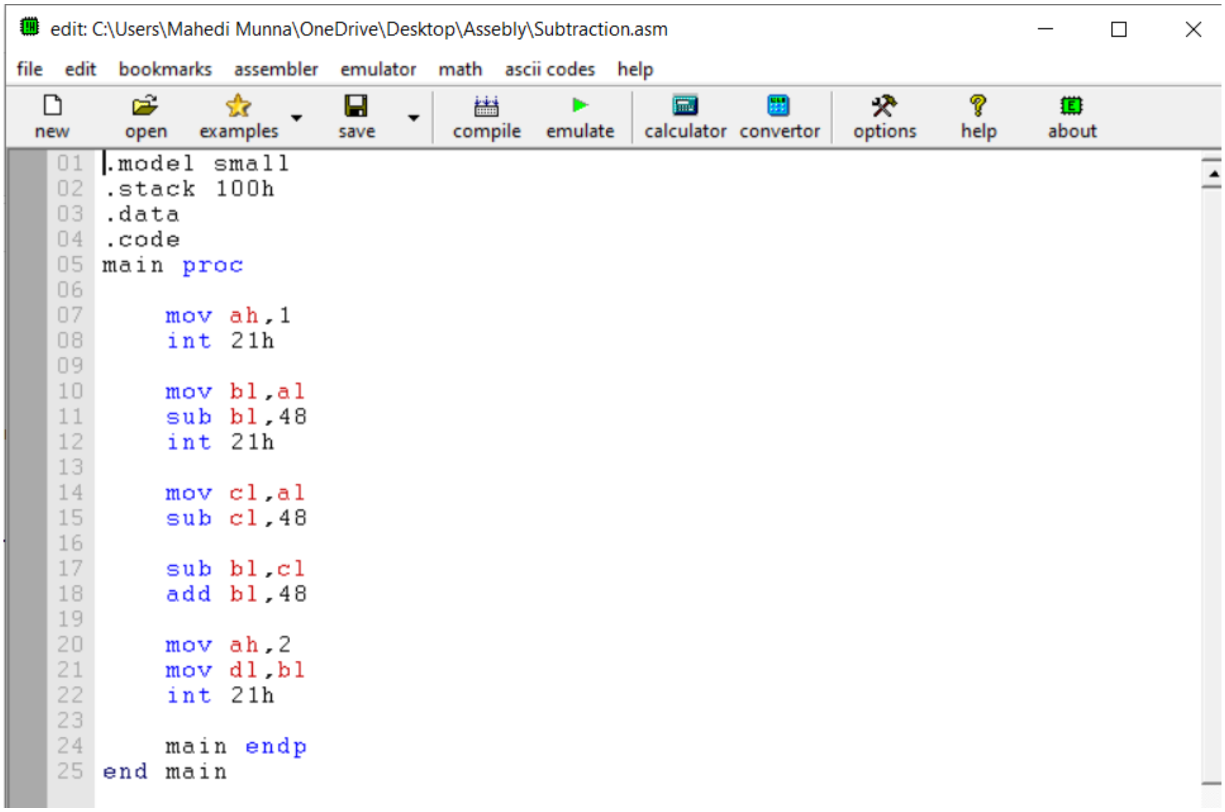
```
.model small
.stack 100h
.data
.code
main proc
    mov ah,1
    int 21h
    mov bl,al   ;data store in al
    sub bl,48

    int 21h
    mov bh,al
    sub bh,48

    int 21h
    mov cl,al

    add bl,bh
    add bl,cl

    mov ah,2
    mov dl,0ah
    int 21h
    mov dl,0dh
    int 21h

    mov dl,bl
    int 21h
    main endp
end main
```

Output:



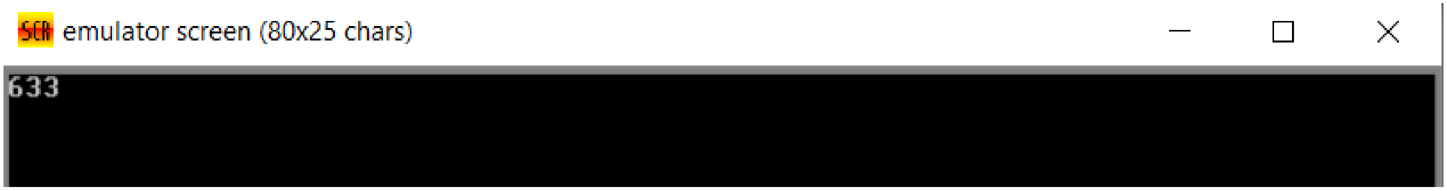
```
123
6
```

4. Subtraction 2 numbers


edit: C:\Users\Mahedi Munna\OneDrive\Desktop\Assebly\Subtraction.asm
file edit bookmarks assembler emulator math ascii codes help
new open examples save compile emulate calculator convertor options help about

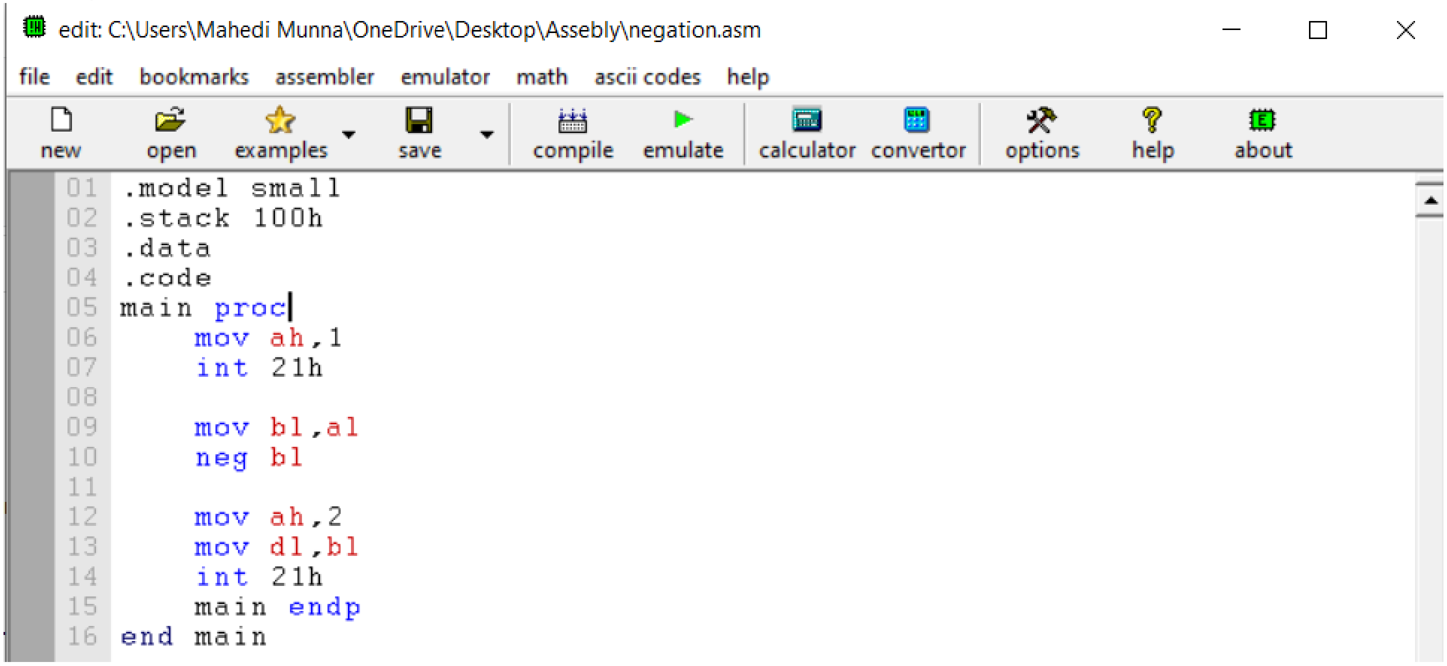
```
.model small
.stack 100h
.data
.code
main proc

    mov ah,1
    int 21h

    mov bl,al
    sub bl,48
    int 21h

    mov cl,al
    sub cl,48

    sub bl,cl
    add bl,48

    mov ah,2
    mov dl,bl
    int 21h

    main endp
end main
```

Output:



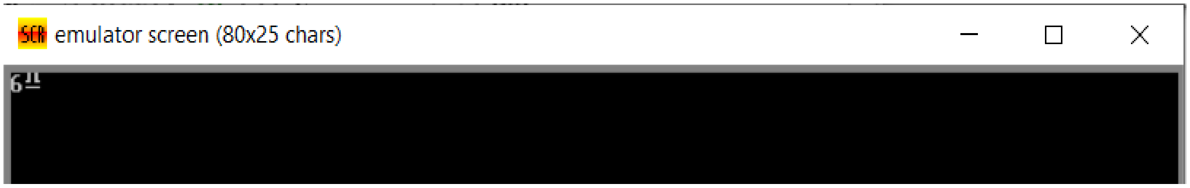
```
633
```

5. Negation


edit: C:\Users\Mahedi Munna\OneDrive\Desktop\Assebly\negation.asm
file edit bookmarks assembler emulator math ascii codes help
new open examples save compile emulate calculator convertor options help about

```
.model small
.stack 100h
.data
.code
main proc
    mov ah,1
    int 21h

    mov bl,al
    neg bl

    mov ah,2
    mov dl,bl
    int 21h
    main endp
end main
```

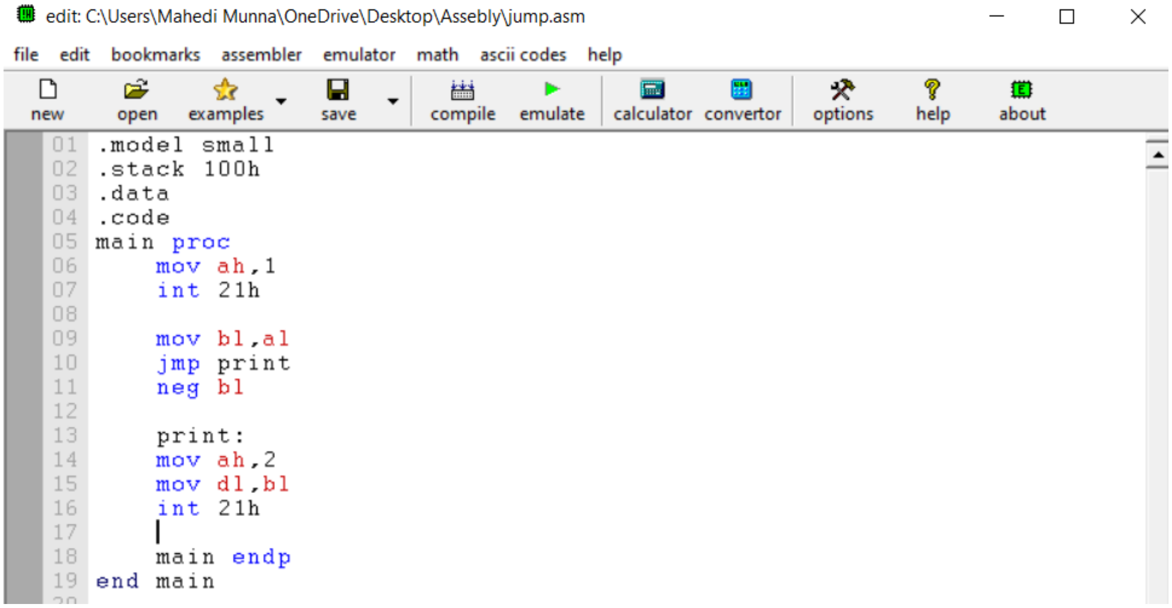
Output:



```
6╩
```

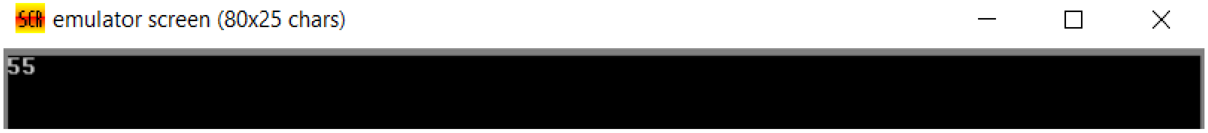
Lab 3

1.Jump



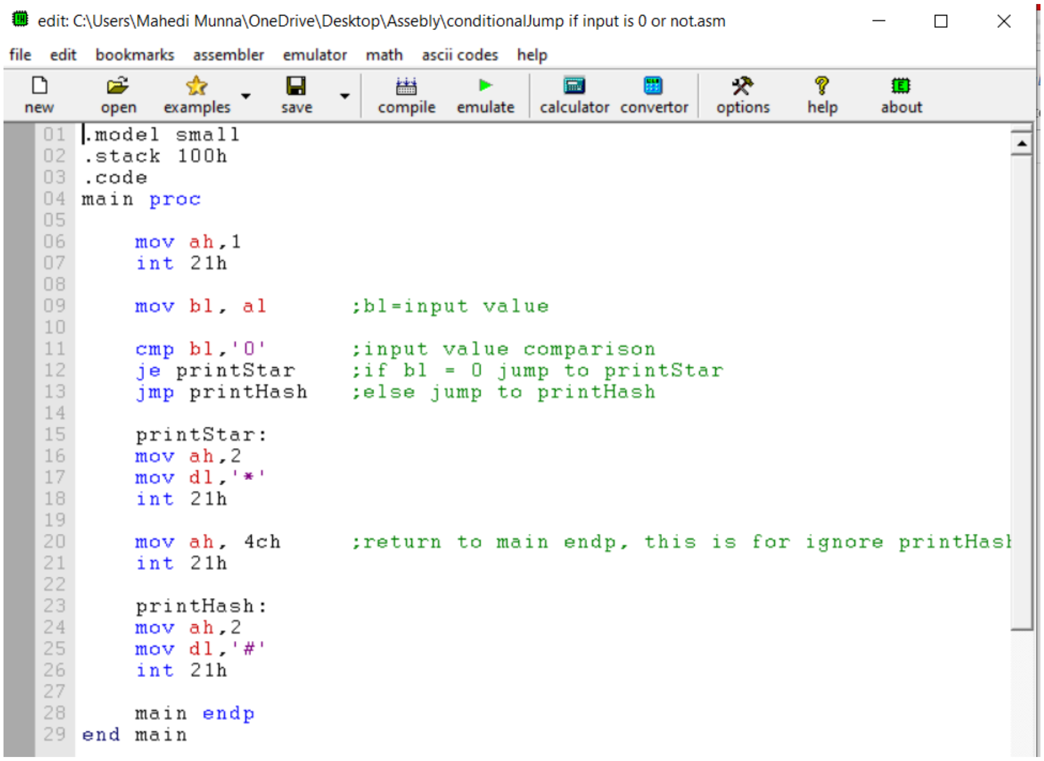
```
.model small
.stack 100h
.data
.code
main proc
     mov ah,1
     int 21h

     mov bl,al
     jmp print
     neg bl

     print:
     mov ah,2
     mov dl,bl
     int 21h

     main endp
end main
```

Output:



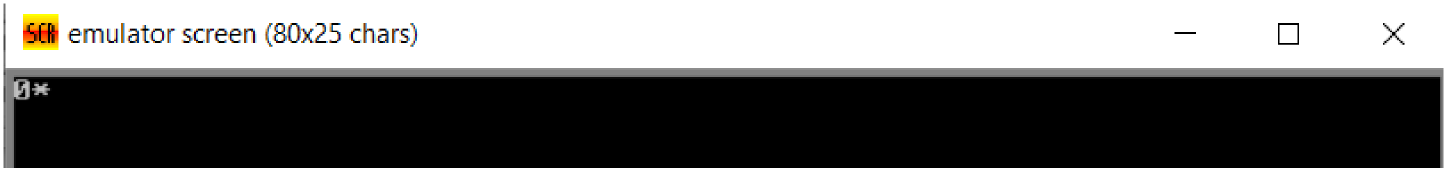
```
55
```

2. Conditional Jump [If zero then print * else #]

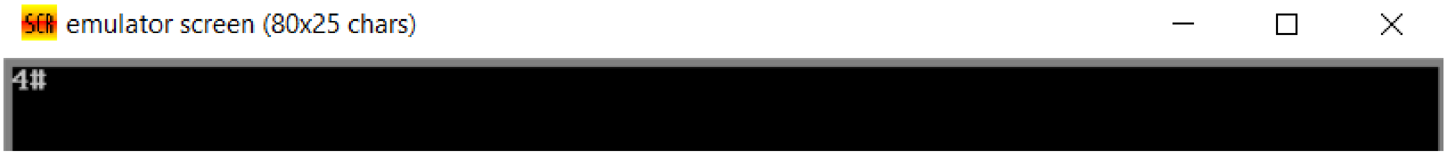


```
.model small
.stack 100h
.code
main proc

    mov ah,1
    int 21h

    mov bl, al      ;bl=input value

    cmp bl,'0'      ;input value comparison
    je printStar    ;if bl = 0 jump to printStar
    jmp printHash   ;else jump to printHash

    printStar:
    mov ah,2
    mov dl,'*'
    int 21h

    mov ah, 4ch     ;return to main endp, this is for ignore printHas

    int 21h

    printHash:
    mov ah,2
    mov dl,'#'
    int 21h

    main endp
end main
```

Output:

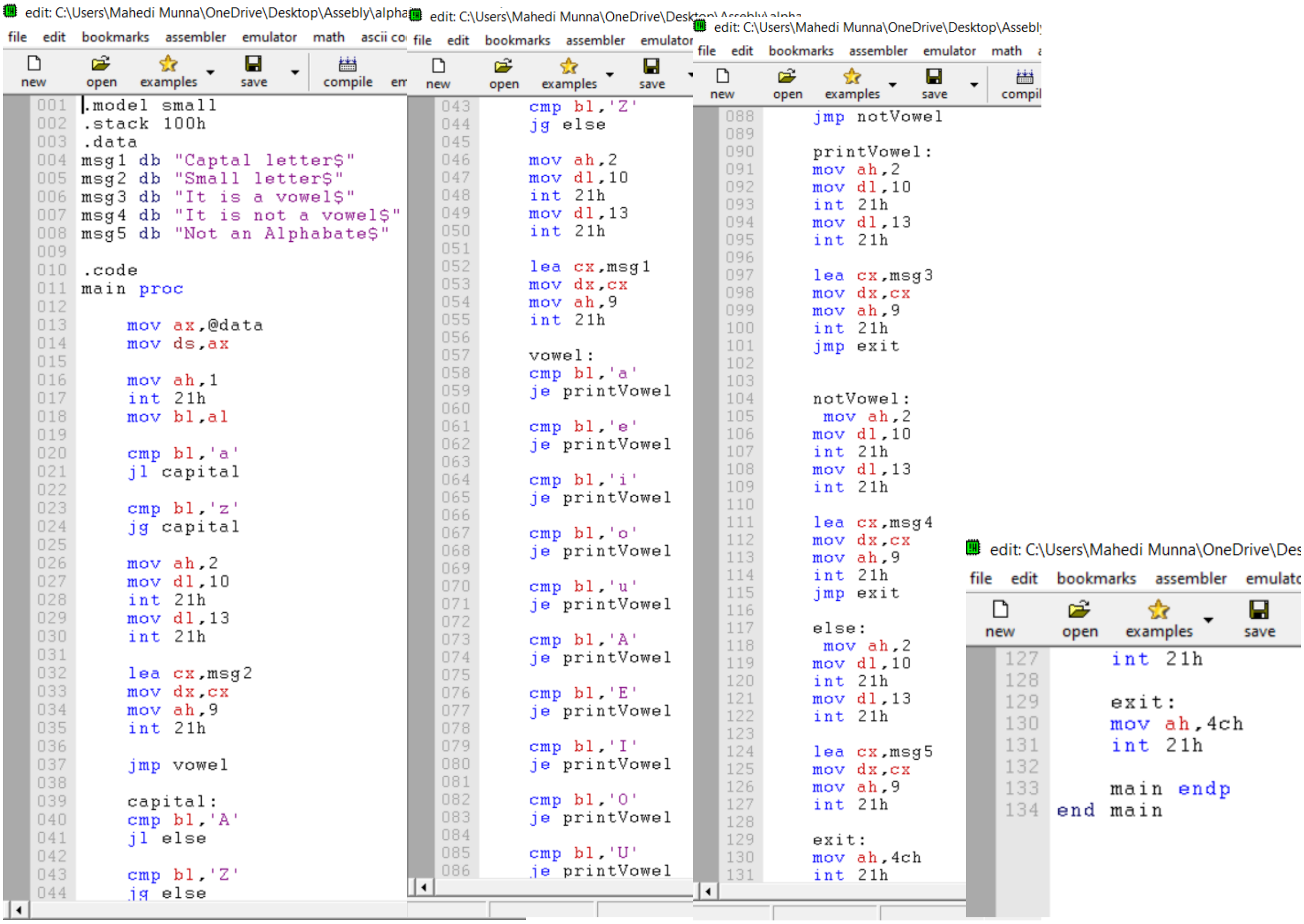


```
0*
```



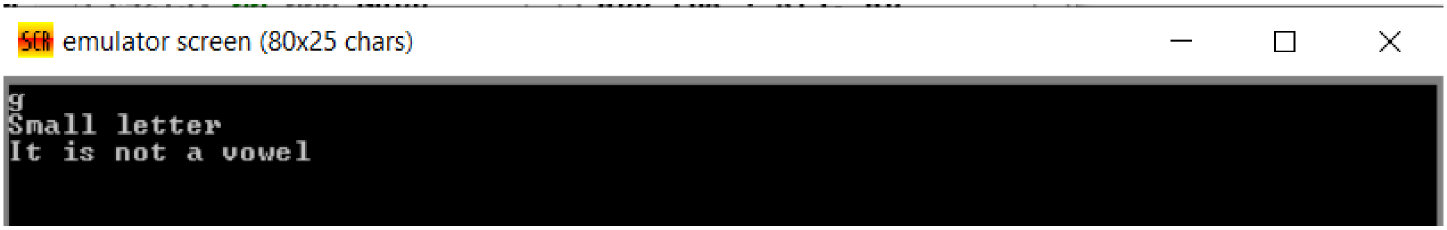
```
4#
```

3. Conditional jump [Capital/small, vowel/not vowel, alphabet/not]

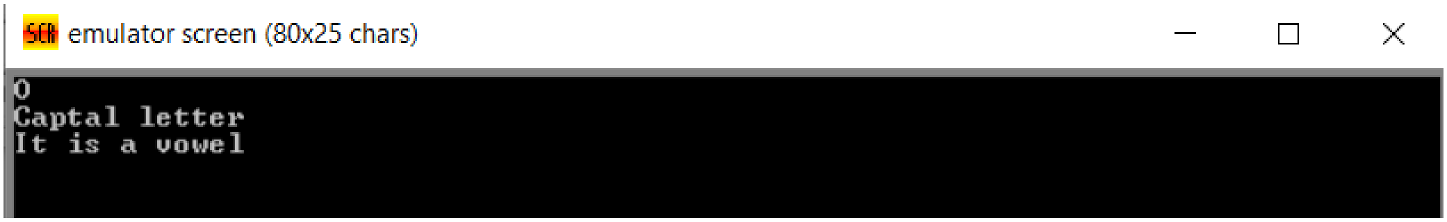

edit: C:\Users\Mahedi Munna\OneDrive\Desktop\Assebly\alpha
file edit bookmarks assembler emulator math ascii co
new open examples save compile em

```
.model small
.stack 100h
.data
msg1 db "Captal letter$"
msg2 db "Small letter$"
msg3 db "It is a vowel$"
msg4 db "It is not a vowel$"
msg5 db "Not an Alphabate$"

.code
main proc

    mov ax,@data
    mov ds,ax

    mov ah,1
    int 21h
    mov bl,al

    cmp bl,'a'
    jl capital

    cmp bl,'z'
    jg capital

    mov ah,2
    mov dl,10
    int 21h
    mov dl,13
    int 21h

    lea cx,msg2
    mov dx,cx
    mov ah,9
    int 21h

    jmp vowel

    capital:
    cmp bl,'A'
    jl else

    cmp bl,'Z'
    jg else

    mov ah,2
    mov dl,10
    int 21h
    mov dl,13
    int 21h

    lea cx,msg1
    mov dx,cx
    mov ah,9
    int 21h

    vowel:
    cmp bl,'a'
    je printVowel

    cmp bl,'e'
    je printVowel

    cmp bl,'i'
    je printVowel

    cmp bl,'o'
    je printVowel

    cmp bl,'u'
    je printVowel

    cmp bl,'A'
    je printVowel

    cmp bl,'E'
    je printVowel

    cmp bl,'I'
    je printVowel

    cmp bl,'O'
    je printVowel

    cmp bl,'U'
    je printVowel
    jmp notVowel

    printVowel:
    mov ah,2
    mov dl,10
    int 21h
    mov dl,13
    int 21h

    lea cx,msg3
    mov dx,cx
    mov ah,9
    int 21h
    jmp exit


    notVowel:
     mov ah,2
    mov dl,10
    int 21h
    mov dl,13
    int 21h

    lea cx,msg4
    mov dx,cx
    mov ah,9
    int 21h
    jmp exit

    else:
     mov ah,2
    mov dl,10
    int 21h
    mov dl,13
    int 21h

    lea cx,msg5
    mov dx,cx
    mov ah,9
    int 21h

    exit:
    mov ah,4ch
    int 21h

    main endp
end main
```

Output:

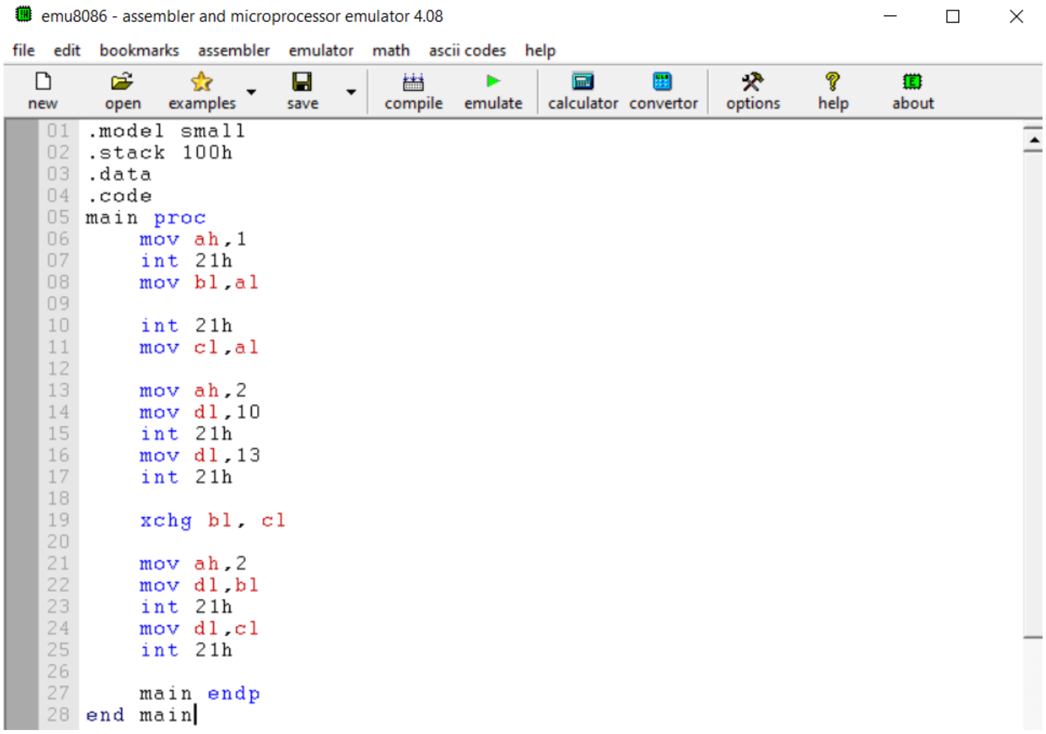


```
g
Small letter
It is not a vowel
```



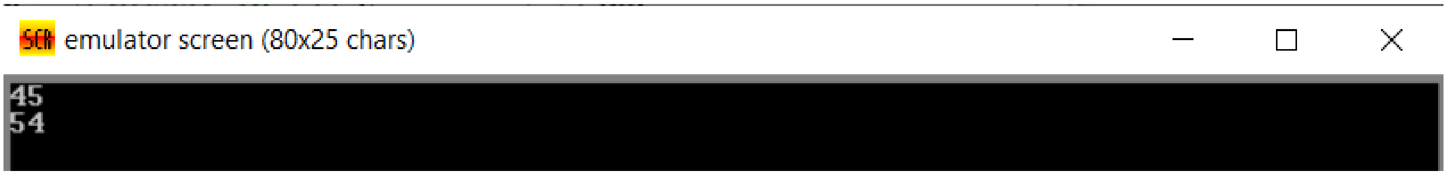
```
O
Captal letter
It is a vowel
```

4. Exchange Number


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 about

```
.model small
.stack 100h
.data
.code
main proc
     mov ah,1
     int 21h
     mov bl,al

     int 21h
     mov cl,al

     mov ah,2
     mov dl,10
     int 21h
     mov dl,13
     int 21h

     xchg bl, cl

     mov ah,2
     mov dl,bl
     int 21h
     mov dl,cl
     int 21h

     main endp
end main
```

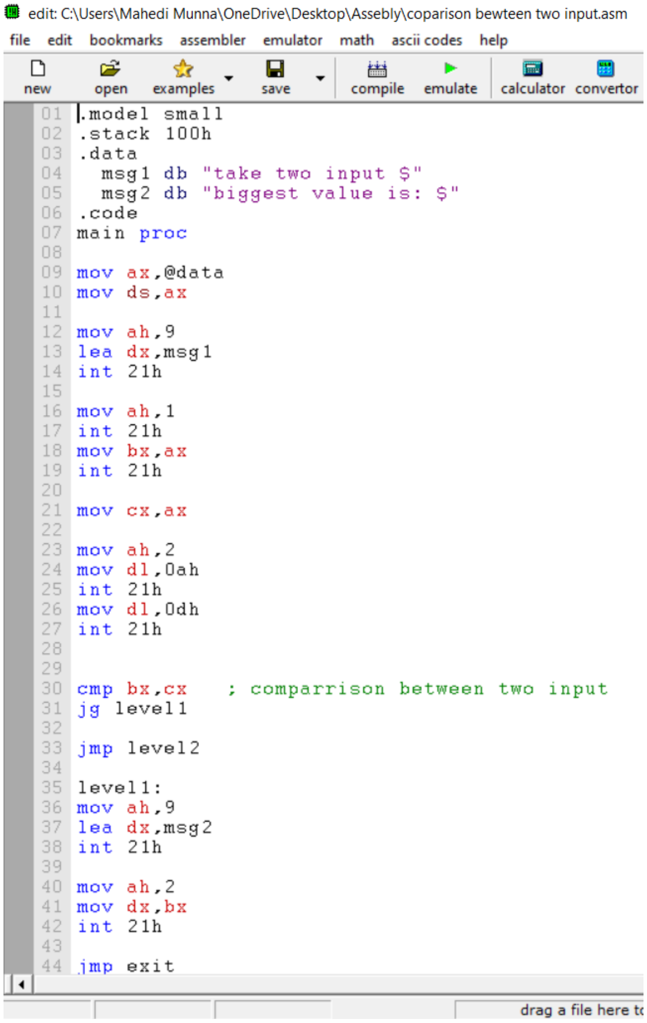
Output:



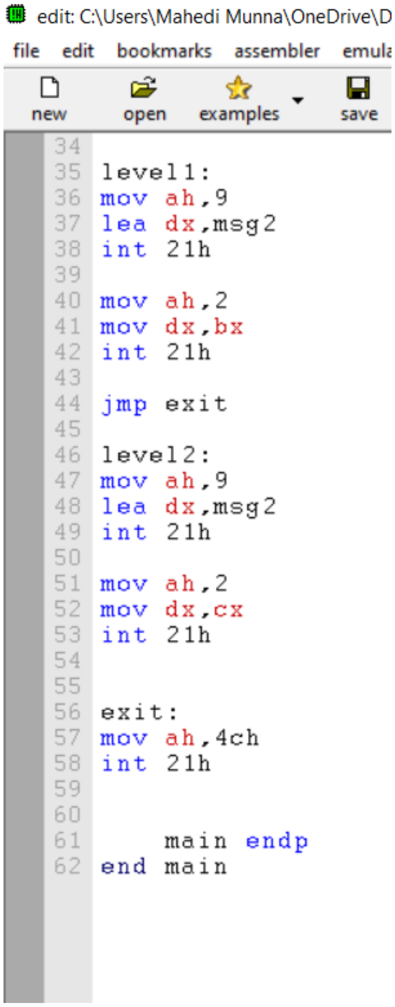
```
45
54
```

Lab 4

1.Compare and print biggest

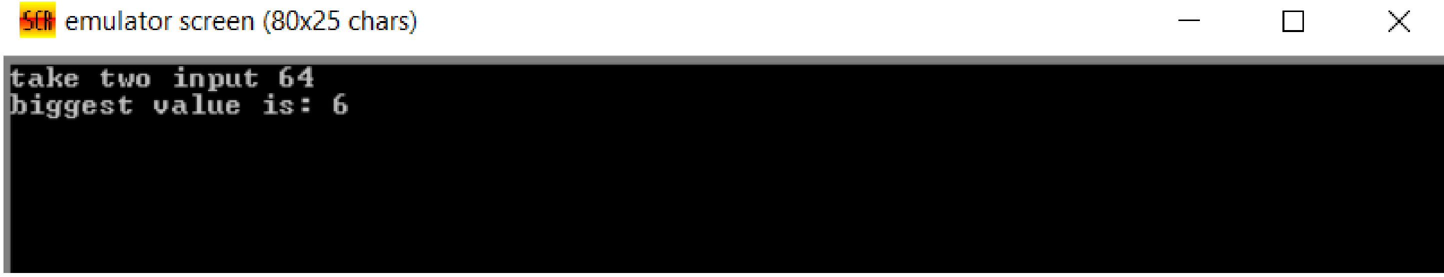

edit: C:\Users\Mahedi Munna\OneDrive\Desktop\Assebly\coparison bewteen two input.asm
file edit bookmarks assembler emulator math ascii codes help
new open examples save compile emulate calculator convertor
drag a file here to

```
.model small
.stack 100h
.data
  msg1 db "take two input $"
  msg2 db "biggest value is: $"
.code
main proc

mov ax,@data
mov ds,ax

mov ah,9
lea dx,msg1
int 21h

mov ah,1
int 21h
mov bx,ax
int 21h

mov cx,ax

mov ah,2
mov dl,0ah
int 21h
mov dl,0dh
int 21h


cmp bx,cx    ; comparrison between two input
jg level1

jmp level2

level1:
mov ah,9
lea dx,msg2
int 21h

mov ah,2
mov dx,bx
int 21h

jmp exit
```


edit: C:\Users\Mahedi Munna\OneDrive\D
file edit bookmarks assembler emula
new open examples save

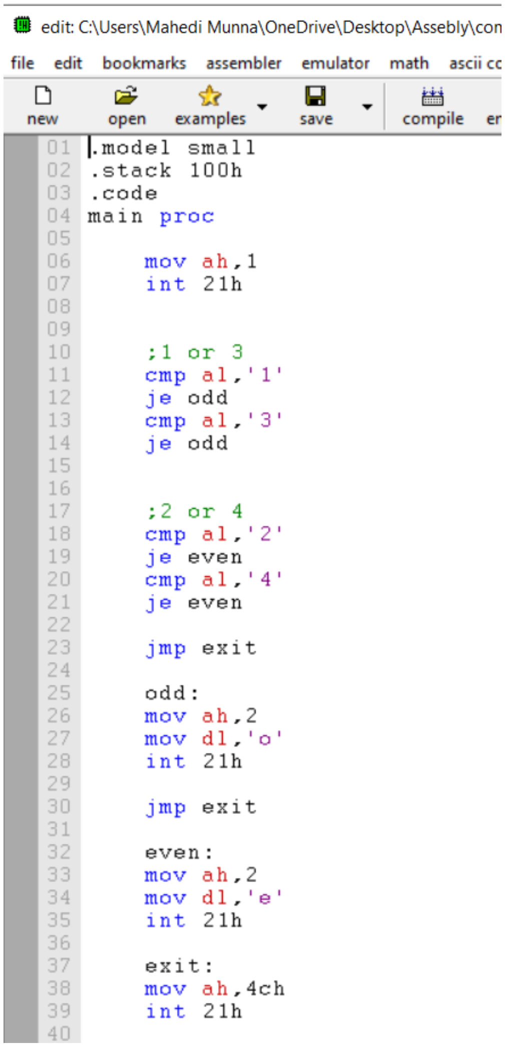
```

level1:
mov ah,9
lea dx,msg2
int 21h

mov ah,2
mov dx,bx
int 21h

jmp exit

level2:
mov ah,9
lea dx,msg2
int 21h

mov ah,2
mov dx,cx
int 21h


exit:
mov ah,4ch
int 21h


     main endp
end main
```

Output:



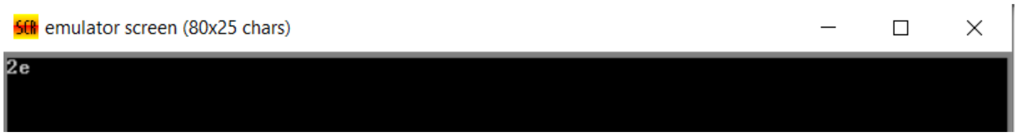
```
take two input 64
biggest value is: 6
```

2. Even or Odd between 1 – 4



```
.model small
.stack 100h
.code
main proc

    mov ah,1
    int 21h


    ;1 or 3
    cmp al,'1'
    je odd
    cmp al,'3'
    je odd


    ;2 or 4
    cmp al,'2'
    je even
    cmp al,'4'
    je even

    jmp exit

    odd:
    mov ah,2
    mov dl,'o'
    int 21h

    jmp exit

    even:
    mov ah,2
    mov dl,'e'
    int 21h

    exit:
    mov ah,4ch
    int 21h
```

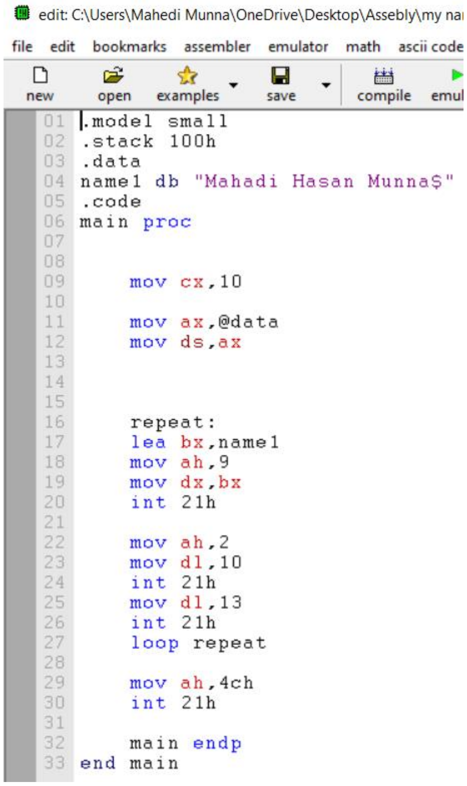
Output:



```
2e
```

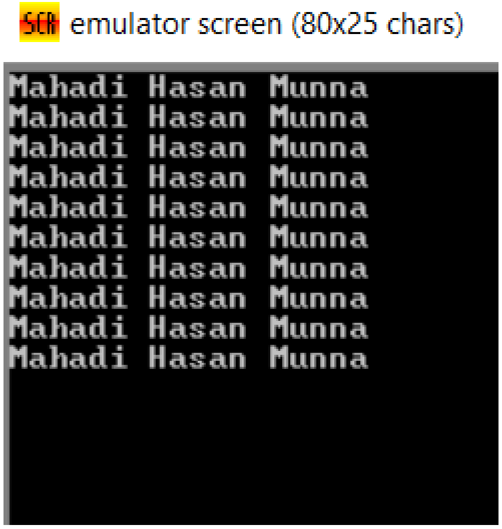
Lab 5

1.Print name 10 times with loop


edit: C:\Users\Mahedi Munna\OneDrive\Desktop\Assebly\my nai
file edit bookmarks assembler emulator math ascii code
new open examples save compile emul

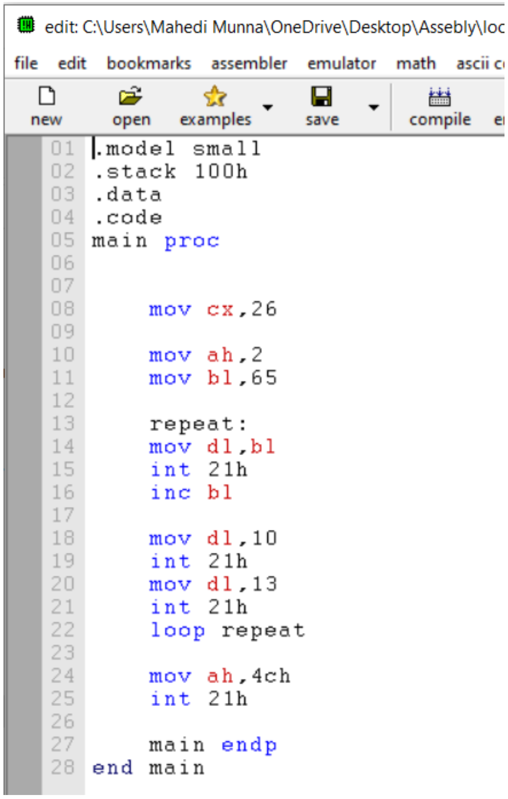
```
.model small
.stack 100h
.data
name1 db "Mahadi Hasan Munna$"
.code
main proc


    mov cx,10

    mov ax,@data
    mov ds,ax


    repeat:
    lea bx,name1
    mov ah,9
    mov dx,bx
    int 21h

    mov ah,2
    mov dl,10
    int 21h
    mov dl,13
    int 21h
    loop repeat

    mov ah,4ch
    int 21h

    main endp
end main
```

Output:



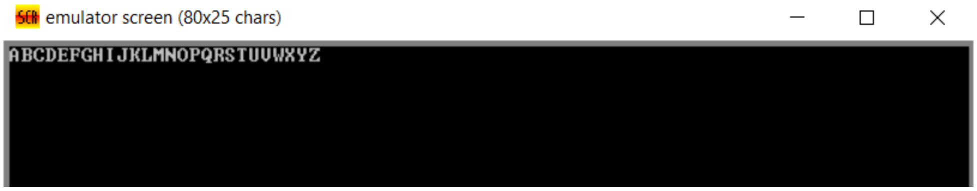
```
Mahadi Hasan Munna
Mahadi Hasan Munna
Mahadi Hasan Munna
Mahadi Hasan Munna
Mahadi Hasan Munna
Mahadi Hasan Munna
Mahadi Hasan Munna
Mahadi Hasan Munna
Mahadi Hasan Munna
Mahadi Hasan Munna
```

2. Print A to Z


edit: C:\Users\Mahedi Munna\OneDrive\Desktop\Assebly\loc
file edit bookmarks assembler emulator math ascii c
new open examples save compile e

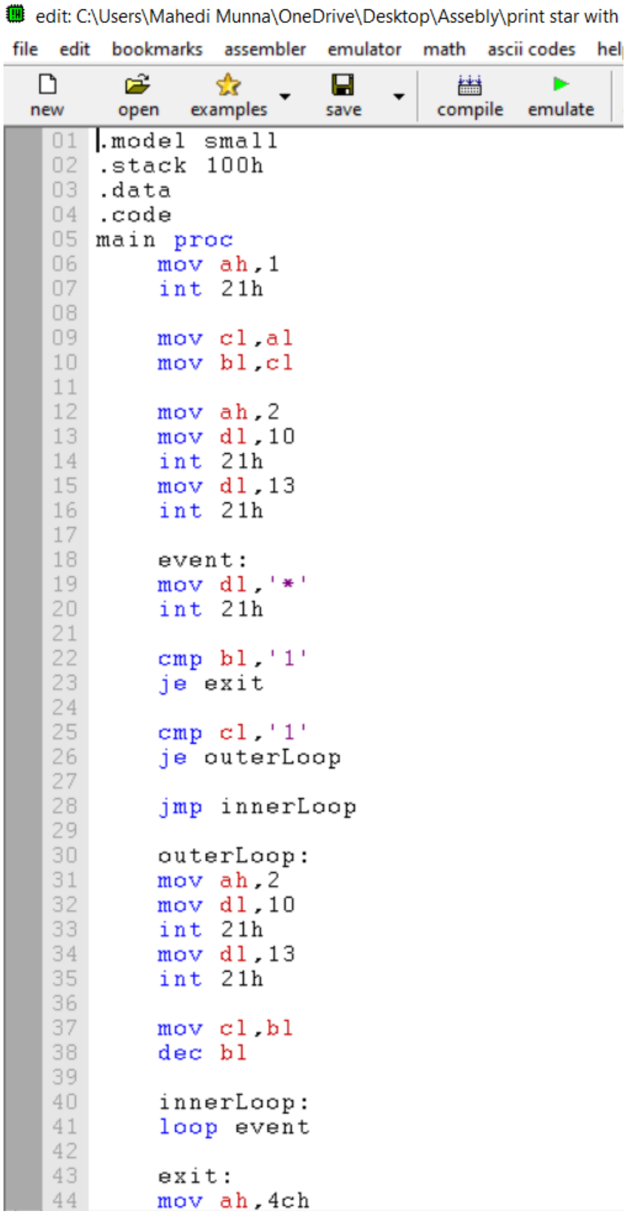
```
.model small
.stack 100h
.data
.code
main proc


    mov cx,26

    mov ah,2
    mov bl,65

    repeat:
    mov dl,bl
    int 21h
    inc bl

    mov dl,10
    int 21h
    mov dl,13
    int 21h
    loop repeat

    mov ah,4ch
    int 21h

    main endp
end main
```

Output:



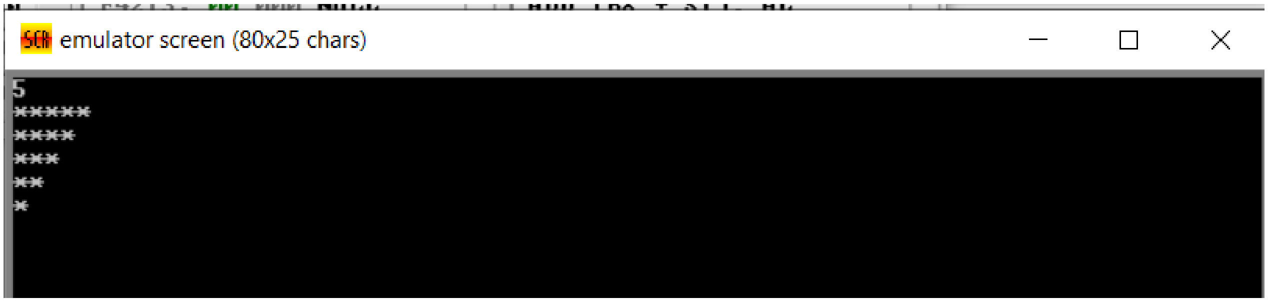
```
ABCDEFGHIJKLMNOPQRSTUVWXYZ
```

3. Print star with decrement



```
.model small
.stack 100h
.data
.code
main proc
    mov ah,1
    int 21h

    mov cl,al
    mov bl,cl

    mov ah,2
    mov dl,10
    int 21h
    mov dl,13
    int 21h

    event:
    mov dl,'*'
    int 21h

    cmp bl,'1'
    je exit

    cmp cl,'1'
    je outerLoop

    jmp innerLoop

    outerLoop:
    mov ah,2
    mov dl,10
    int 21h
    mov dl,13
    int 21h

    mov cl,bl
    dec bl

    innerLoop:
    loop event

    exit:
    mov ah,4ch
```

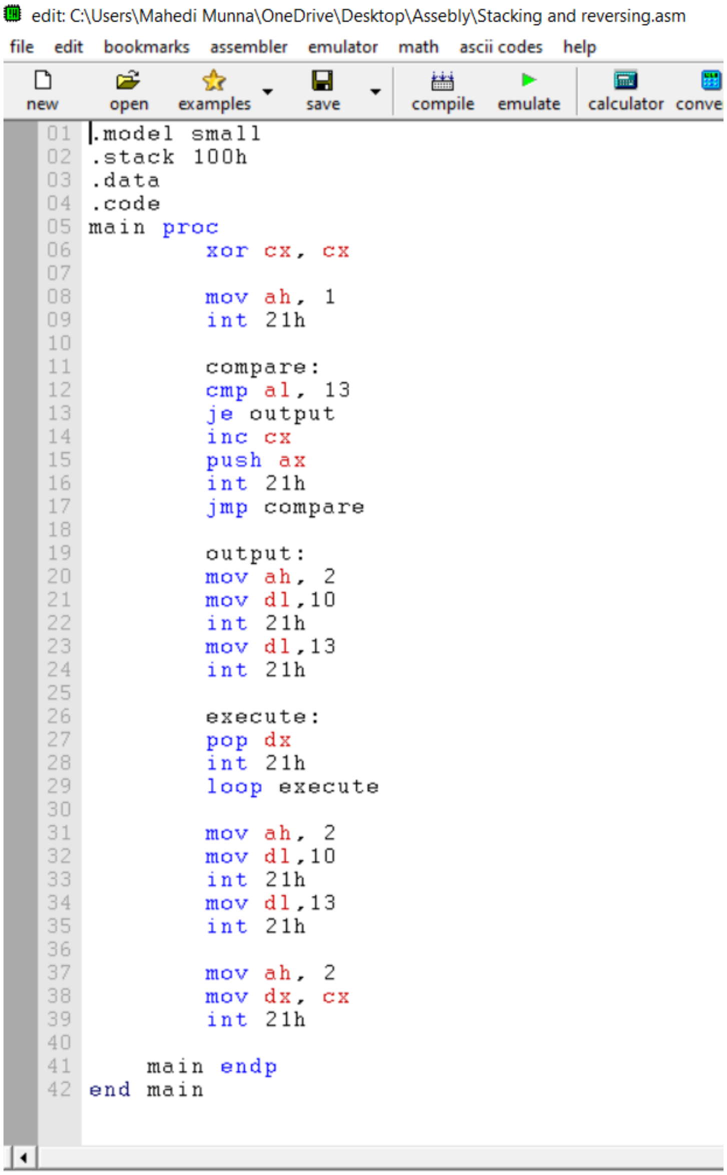
Output:



```
5
*****
****
***
**
*
```

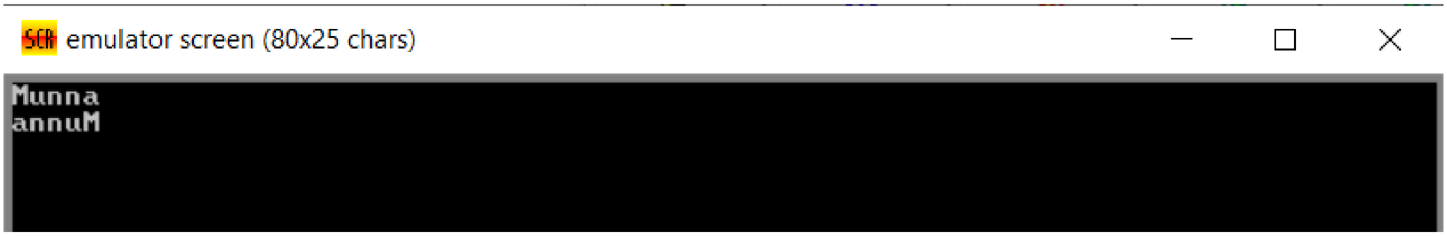
Lab 6

1.Reverse a string



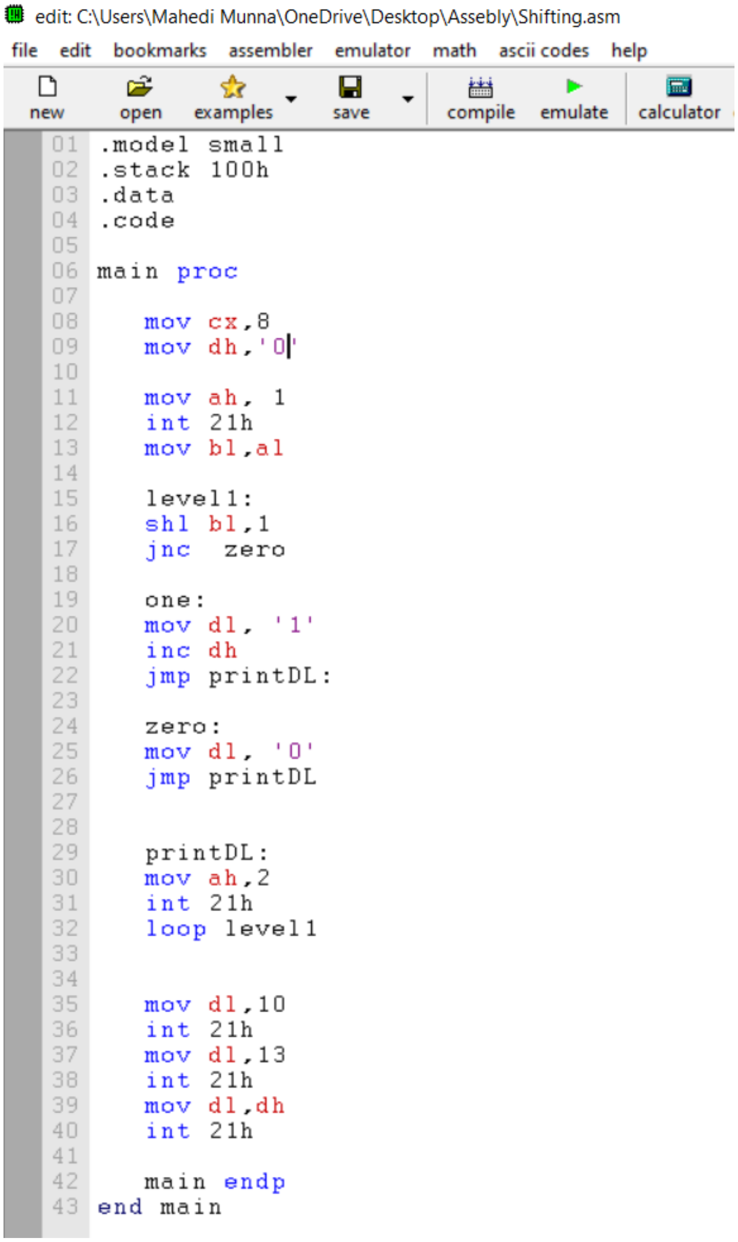
```
.model small
.stack 100h
.data
.code
main proc
        xor cx, cx

        mov ah, 1
        int 21h

        compare:
        cmp al, 13
        je output
        inc cx
        push ax
        int 21h
        jmp compare

        output:
        mov ah, 2
        mov dl,10
        int 21h
        mov dl,13
        int 21h

        execute:
        pop dx
        int 21h
        loop execute

        mov ah, 2
        mov dl,10
        int 21h
        mov dl,13
        int 21h

        mov ah, 2
        mov dx, cx
        int 21h

    main endp
end main
```

Output:



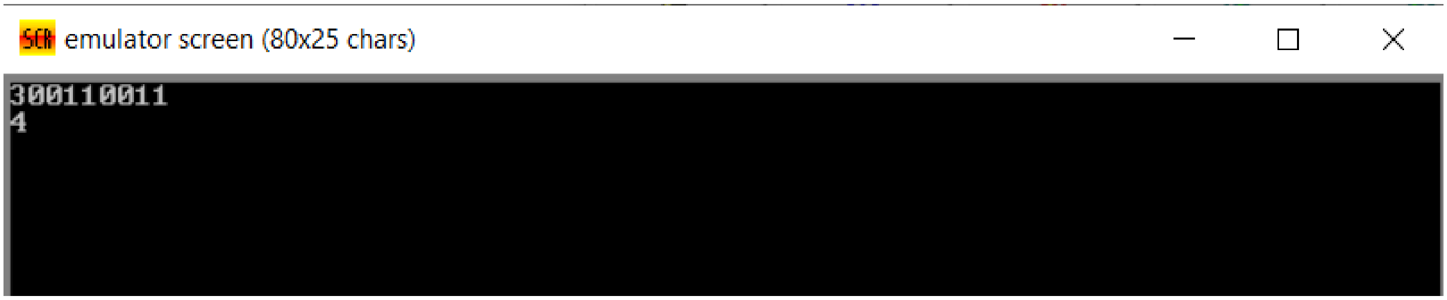
```
Munna
annuM
```

2. Count number of 1



```
.model small
.stack 100h
.data
.code

main proc

    mov cx,8
    mov dh,'0'

    mov ah, 1
    int 21h
    mov bl,al

    level1:
    shl bl,1
    jnc  zero

    one:
    mov dl, '1'
    inc dh
    jmp printDL:

    zero:
    mov dl, '0'
    jmp printDL


    printDL:
    mov ah,2
    int 21h
    loop level1


    mov dl,10
    int 21h
    mov dl,13
    int 21h
    mov dl,dh
    int 21h

    main endp
end main
```

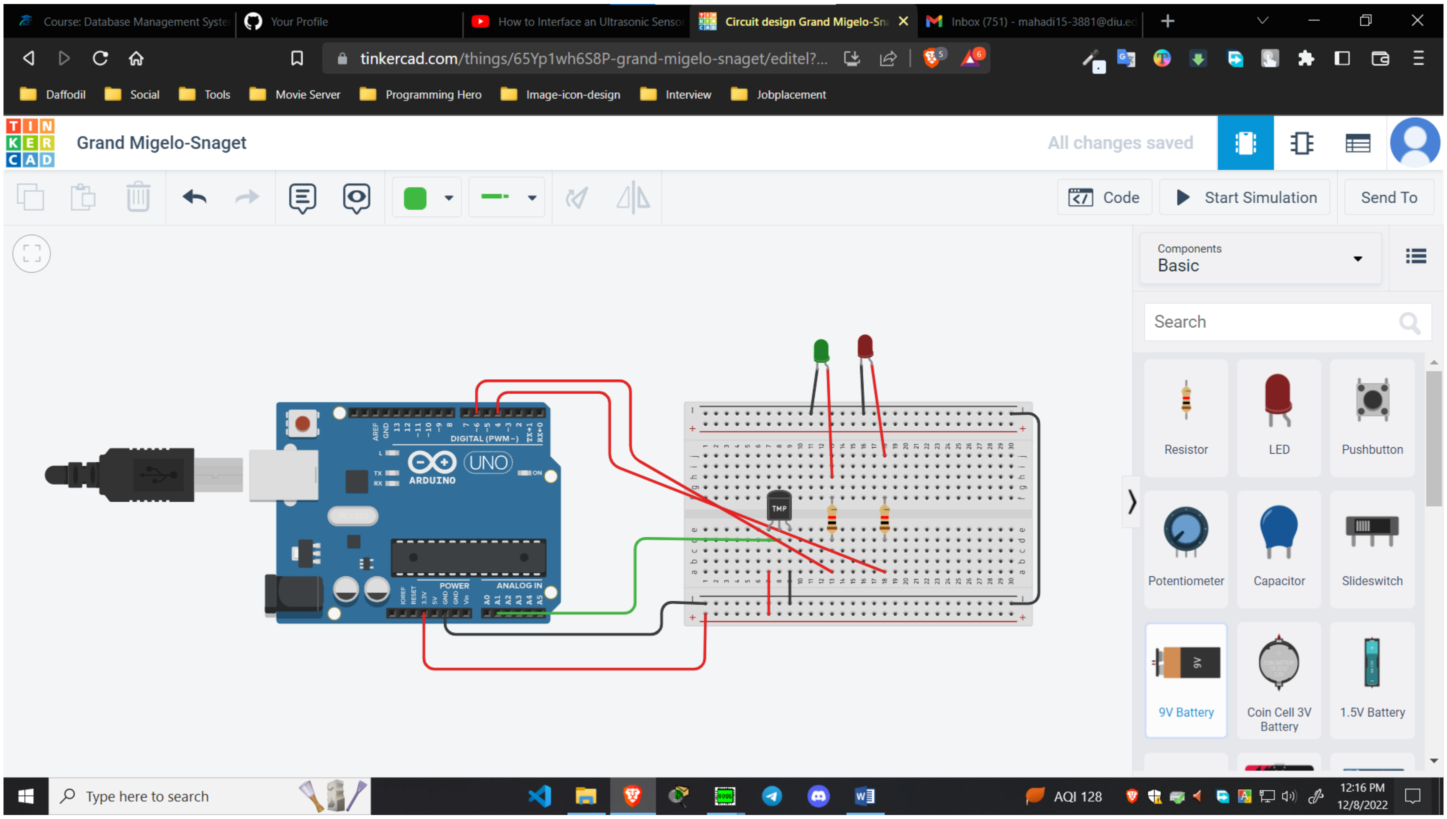
Output:



```
300110011
4
```

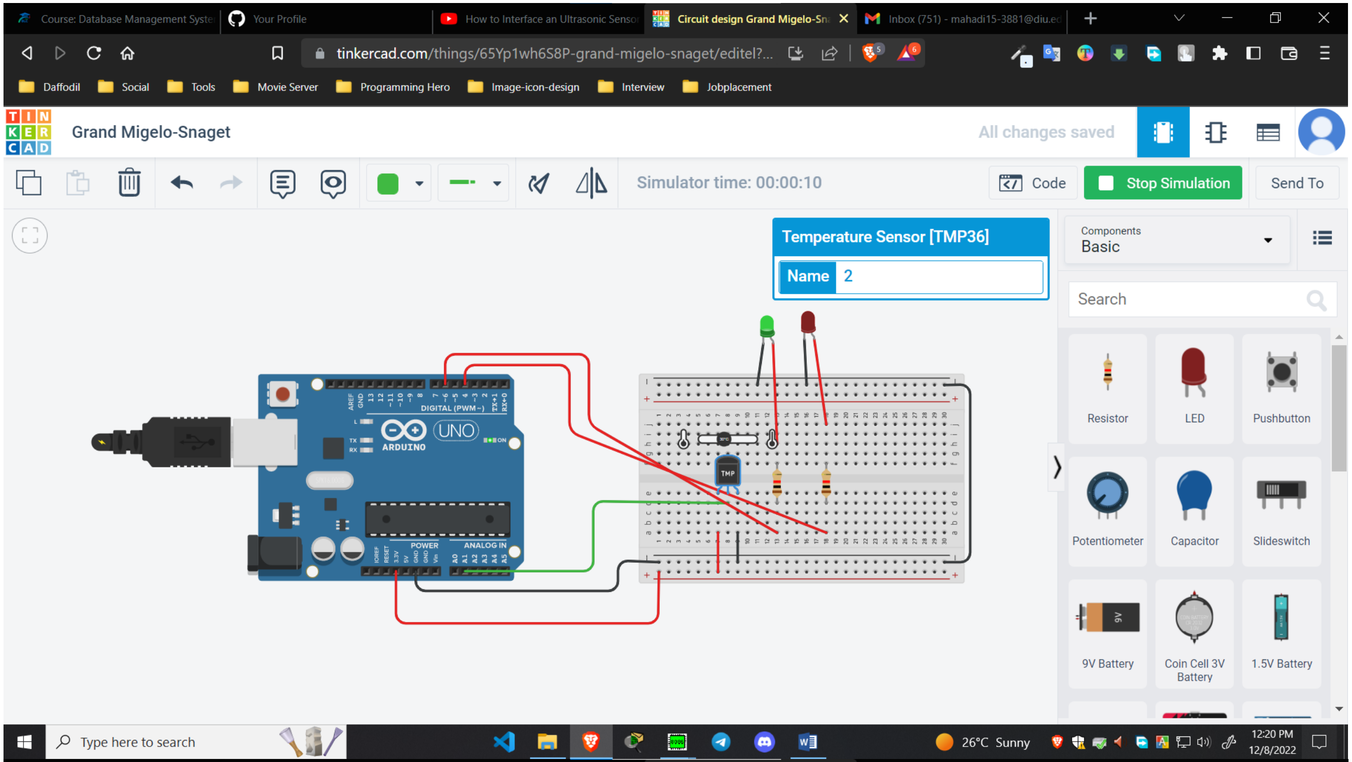
Lab 7

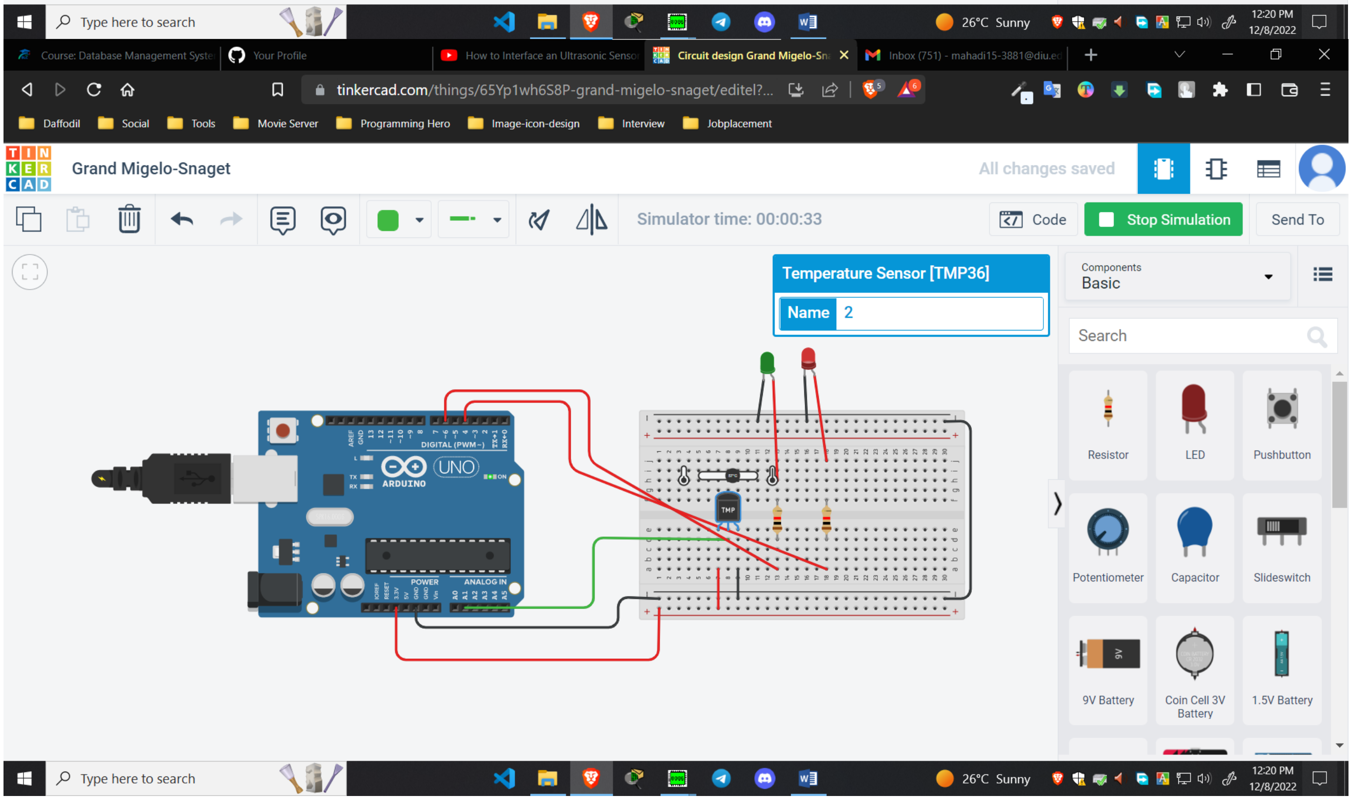
1. Temperature Sensor:



Code:

```
void setup()
{
 pinMode(A1, INPUT);
 pinMode(6, OUTPUT);
 pinMode(4, OUTPUT);
 Serial.begin(9600);
}
void loop()
{
        float value = analogRead(A1);
        float voltage = value * (5/1024.0);
        float temp = (voltage - 0.5) * 100;
        Serial.print("Temp value: ");
        Serial.println(temp);
        if(temp>10) {
                if(temp<50) {
                        digitalWrite(6, HIGH);
                        digitalWrite(4, LOW);
                }
                else {
                        digitalWrite(4,HIGH);
                        digitalWrite(6,LOW);
        }
        }
        else{
        digitalWrite(4,HIGH);
        digitalWrite(6,LOW);
        }
        delay(100);
}
```


Course: Database Management Syste
Your Profile
How to Interface an Ultrasonic Senso
Circuit design Grand Migelo-Sn
Inbox (751) - mahadi15-3881@diu.ed
tinkercad.com/things/65Yp1wh6S8P-grand-migelo-snaget/editel?...
Daffodil
Social
Tools
Movie Server
Programming Hero
Image-icon-design
Interview
Jobplacement
Grand Migelo-Snaget
All changes saved
Simulator time: 00:00:10
Code
Stop Simulation
Send To
Temperature Sensor [TMP36]
Name 2
Components
Basic
Search
Resistor
LED
Pushbutton
Potentiometer
Capacitor
Slideswitch
9V Battery
Coin Cell 3V Battery
1.5V Battery
Type here to search
26°C Sunny
12:20 PM
12/8/2022


Course: Database Management Syste
Your Profile
How to Interface an Ultrasonic Senso
Circuit design Grand Migelo-Sn
Inbox (751) - mahadi15-3881@diu.ed
tinkercad.com/things/65Yp1wh6S8P-grand-migelo-snaget/editel?...
Daffodil
Social
Tools
Movie Server
Programming Hero
Image-icon-design
Interview
Jobplacement
Grand Migelo-Snaget
All changes saved
Simulator time: 00:00:33
Code
Stop Simulation
Send To
Temperature Sensor [TMP36]
Name 2
Components
Basic
Search
Resistor
LED
Pushbutton
Potentiometer
Capacitor
Slideswitch
9V Battery
Coin Cell 3V Battery
1.5V Battery
Type here to search
26°C Sunny
12:20 PM
12/8/2022