****

**CSC221- COMPUTER ORGANIZATION & ASSEMBLY LANGUAGE (LAB)**

Spring 2023- (Session ID: 372302022)

Name: MUHAMMAD HUZAIFA

Roll No: 15903

Session Day: WEDNESDAY (3-5)

**Lab Task 1:** We learned how to print a letter on screen using constant value or by ASCII. You are required to write a code to display your name on the screen.

.model small

.stack 100h

.data

.code

Main proc

Mov dl,'H'

Mov ah,2

INT 21h

Mov dl,'U'

Mov ah,2

INT 21h

Mov dl,'Z'

Mov ah,2

INT 21h

Mov dl,'A'

Mov ah,2

INT 21h

Mov dl,'I'

Mov ah,2

INT 21h

Mov dl,'F'

Mov ah,2

INT 21h

Mov dl,'A'

Mov ah,2

INT 21h

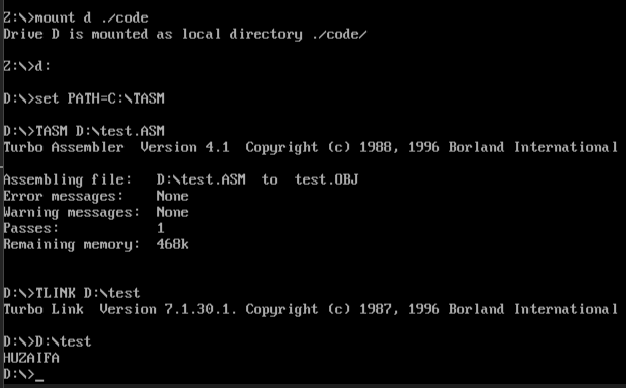
Mov ah,4ch

INT 21h

Main endp

End Main

**OUTPUT:**

****

**Lab Task 2:** We learned how to add take input from users, add the received numbers, and print on screen using Assembly language. You are required to write a code to display the result on the next line instead of the same line using ASCII code.

dosseg

.model small

.stack 100h

.data

.code

Main proc

Mov ah,1

INT 21h

mov bl,al

mov al,1

INT 21h

add bl,al

sub bl,48

mov dl,10

mov ah,2

INT 21h

mov dl,bl

mov ah,2

INT 21h

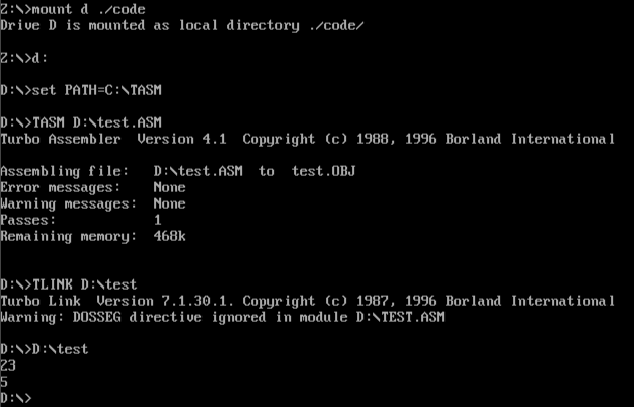
mov ah,4ch

INT 21h

Main endp

End Main

**OUTPUT:**



**Lab Task 3:** We practiced getting user input of the capital alphabet and converting it into the small alphabet. You must write a code that accepts user input of a small alphabet and converts it into the capital alphabet.

dosseg

.model small

.stack 100h

.data

.code

Main proc

Mov ah,1

INT 21h

Mov dl,al

sub dl,32

Mov ah,2

INT 21h

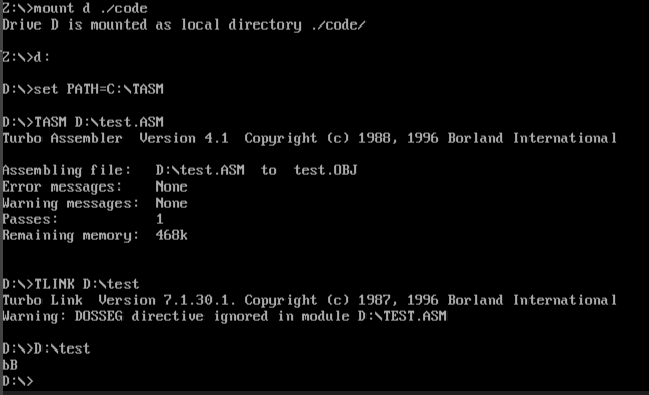
mov ah,4ch

INT 21h

Main endp

End Main

**OUTPUT:**

****

**Lab Task 4:**

Using variables, create a following output:

S

H

A

H

A

B

A

Z

A

M

You can use your name to display in such a manner but using variables.

dosseg

.model small

.data

m1 db "M$"

m2 db " U$"

m3 db " H$"

m4 db " A$"

m5 db " M$"

m6 db " M$"

m7 db " A$"

m8 db " D$"

m9 db " H$"

m10 db " U$"

m11 db " Z$"

m12 db " A$"

m13 db " I$"

m14 db " F$"

m15 db " A$"

.code

main proc

mov ax,@data

mov ds, ax

mov ah,9

lea dx,m1

int 21h

mov dx,10

mov ah,2

int 21h

mov ah,9

lea dx,m2

int 21h

mov dx,10

mov ah,2

int 21h

mov ah,9

lea dx,m3

int 21h

mov dx,10

mov ah,2

int 21h

mov ah,9

lea dx,m4

int 21h

mov dx,10

mov ah,2

int 21h

mov ah,9

lea dx,m5

int 21h

mov dx,10

mov ah,2

int 21h

mov ah,9

lea dx,m6

int 21h

mov dx,10

mov ah,2

int 21h

mov ah,9

lea dx,m7

int 21h

mov dx,10

mov ah,2

int 21h

mov ah,9

lea dx,m8

int 21h

mov dx,10

mov ah,2

int 21h

mov ah,9

lea dx,m9

int 21h

mov dx,10

mov ah,2

int 21h

mov ah,9

lea dx,m10

int 21h

mov dx,10

mov ah,2

int 21h

mov ah,9

lea dx,m11

int 21h

mov dx,10

mov ah,2

int 21h

mov ah,9

lea dx,m12

int 21h

mov dx,10

mov ah,2

int 21h

mov ah,9

lea dx,m13

int 21h

mov dx,10

mov ah,2

int 21h

mov ah,9

lea dx,m14

int 21h

mov dx,10

mov ah,2

int 21h

mov ah,9

lea dx,m15

int 21h

mov dx,10

mov ah,2

int 21h

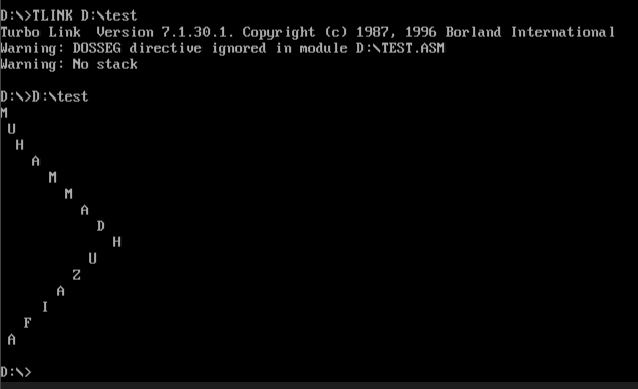
mov ah,4ch

int 21h

main endp

end main

**OUTPUT:**

****

**Lab Task 5:** Using loop do the following:

1. Print the small alphabet from a-z.

2. Print capital alphabet in reverse order from Z-A

3. Print Odd numbers from 0 to 9.

4. Print Even numbers from 0 to 9

5. Print your Roll number 5 times.

**1**

dosseg

.model small

.data

.code

main proc

mov cx, 26

mov dx,97

l1:

mov ah,2

int 21h

Add dx,1

Loop l1

mov ah,4ch

int 21h

main endp

**OUTPUT:**

****

**2**

.model small

.stack 100h

.data

.code

main proc

mov cx,26

mov dx,90

l1:

mov ah,2

int 21h

sub dx,1

loop l1

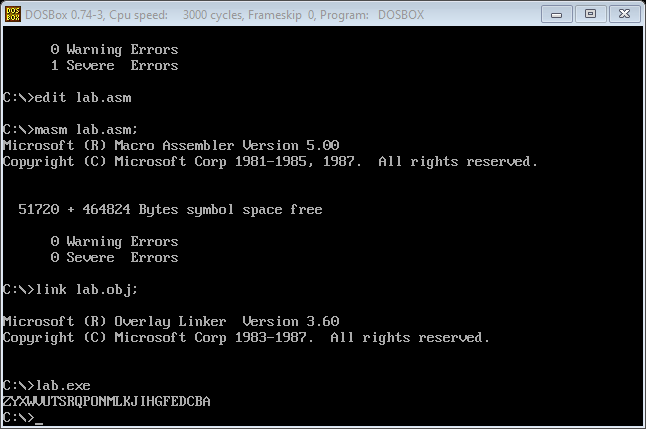
mov ah,4ch

int 21h

main endp

end main

**OUTPUT:**



**3**

dosseg

.model small

.data

.code

main proc

mov cx,5

mov dx,49

l1:

mov ah,2

int 21h

Add dx,2

loop l1

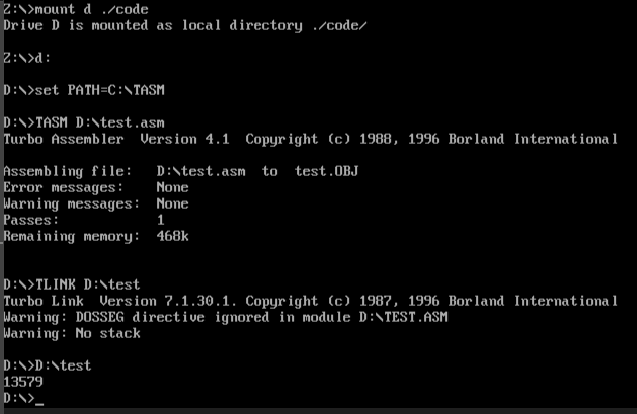
mov ah,4ch

int 21h

main endp

end main

**OUTPUT:**

****

**4**

dosseg

.model small

.data

.code

main proc

mov cx,5

mov dx,48

l1:

mov ah,2

int 21h

Add dx, 2

Loop l1

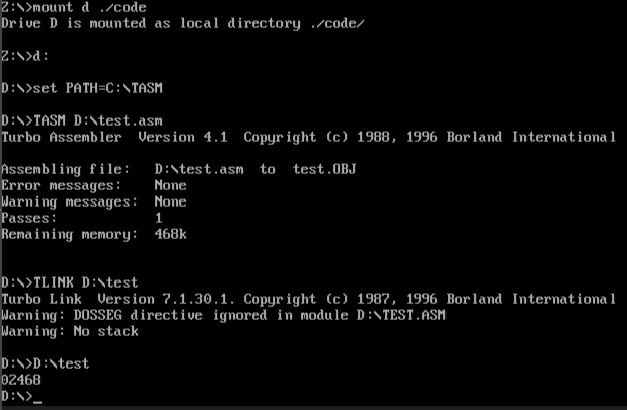
mov ah,4ch

int 21h

main endp

end main

**OUTPUT:**

****

**5**

dosseg

.model small

.data

var1 db "15903 $"

.code

main proc

mov cx,5

mov dx,@data

l1:

mov ax,@data

mov ds,ax

mov dx,offset var1

mov ah,9

int 21h

mov dx,10

mov ah,2

int 21h

mov ah,2

int 21h

loop l1

mov ah,4ch

int 21h

main endp

end main

**OUTPUT:**



**Lab Task 6: Using a nested loop, create the following patterns.**

1.

\* \* \* \* \*

\* \* \* \*

\* \* \*

\* \*

\*

2.

\*

\* \*

\* \* \*

\* \* \* \*

\* \* \* \* \*

\* \* \* \*

\* \* \*

\* \*

\*

3.

Your name

your roll no

your roll no

your name

your roll no

your roll no

1

dosseg

.model small

.stack 100h

.data

.code

main proc

mov ax,@data

mov ds,ax

mov bx, 1

mov cx, 5

L1:

push cx

mov bx, cx

L2:

Mov dl, '\*'

mov ah,2

int 21h

loop L2

mov dl,10

mov ah, 2

int 21h

mov dl,13

mov ah, 2

int 21h

inc bl

pop cx

loop L1

mov ah,4ch

int 21h

main endp

end main

**OUTPUT:**



2

dosseg

.model small

.stack 100h

.data

.code

main proc

mov ax,@data

mov ds,ax

mov bx, 1

mov cx, 5

L1:

push cx

mov cx, bx

L2:

Mov dl, '\*'

mov ah,2

int 21h

loop L2

mov dl,10

mov ah, 2

int 21h

mov dl,13

mov ah, 2

int 21h

inc bl

pop cx

loop L1

mov bx, 1

mov cx, 5

L3:

push cx

mov bx, cx

L4:

Mov dl, '\*'

mov ah,2

int 21h

loop L4

mov dl,10

mov ah, 2

int 21h

mov dl,13

mov ah, 2

int 21h

inc bl

pop cx

loop L3

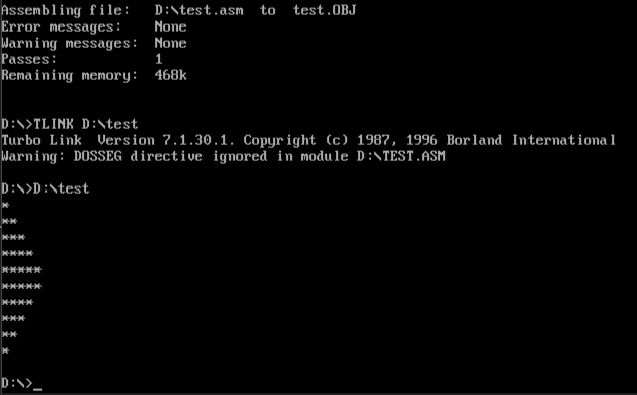
mov ah,4ch

int 21h

main endp

end main

**OUTPUT:**



3

dosseg

.model small

.stack 100h

.data

vr1 db 'MUHAMMAD HUZAIFA$'

vr2 db '       15903$'

.code

main proc

mov ax,@data

mov ds,ax

mov cx,2

Loop1:

lea dx,vr1

mov ah,9

int 21h

push cx

mov cx,2

Loop2:

mov dl,10

mov ah,2

int 21h

lea dx,vr2

mov ah,9

int 21h

inc bl

loop Loop2

mov dl,10

mov ah,2

int 21h

inc bl

pop cx

loop Loop1

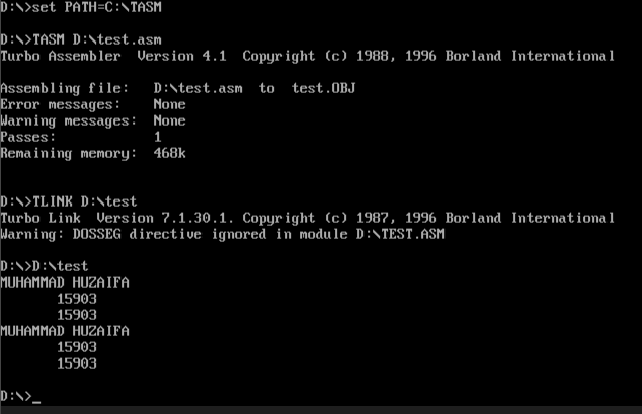
mov ah,4ch

int 21h

main endp

end main

**OUTPUT:**



**Lab Task 7:** Write an assembly language code to collect 2 digits using an array and compare them. If the numbers are equal then print EQUAL, if the numbers are not equal then print NOT EQUAL.

.model small

.stack 100h

.data

msg db 'enter 2 digits :$'

v1 db 'equal$'

v2 db 'not equal$'

arr db 3 dup('$')

.code

main proc

mov ax,@data

mov ds,ax

mov dx,offset msg

mov ah,9

int 21h

lea si,arr

mov ah,1

int 21h

mov cl,al

mov ah,2

int 21h

mov ah,1

int 21h

mov dl,al

cmp dl,cl

je print

mov dx,offset v2

mov ah,9

int 21h

mov ah,4ch

int 21h

print:

mov dx,offset v1

mov ah,9

int 21h

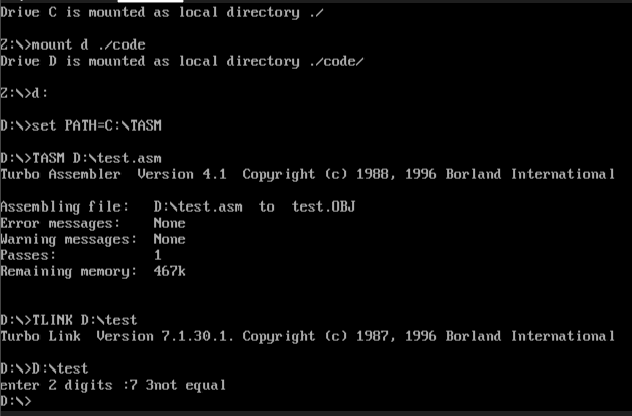
mov ah, 4ch

int 21h

main endp

end main

**OUTPUT:**



**Lab Task 8:**

1. Write an assembly language code to collect 2 digits and compare them. If the first number is greater than the second number then print "Number 1 is Greater than Number 2", if number 1 is less than number 2 then print "Number 1 is less than number 2".  are GREATER or LESS than others.
2. Write a program to read a character from the user and display that character 100 times on the next line using a loop.

1

dosseg

.model small

.stack 100h

.data

huzaifa1 db "NUMBER 1 IS GREATER THAN NUMBER 2$"

huzaifa2 db "NUMBER 1 IS LESS THAN NUMBER 2$"

.code

main proc

mov ax,@data

mov ds,ax

mov dl,'4'

mov ah,1

int 21h

cmp al,dl

je l1

mov dx,offset huzaifa2

mov ah,9

int 21h

mov dl,10

mov ah,2

int 21h

l1:

mov dx,offset huzaifa1

mov ah,9

int 21h

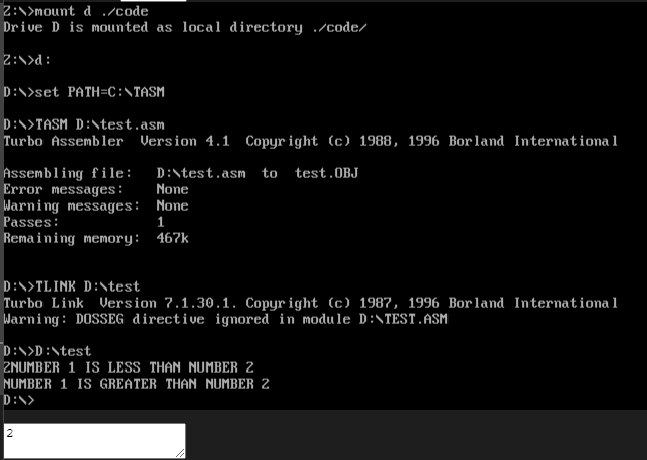
mov ah,4ch

int 21h

main endp

end main

**OUTPUT:**



2

dosseg

.model small

.stack 100h

.data

.code

main proc

mov ah,1

int 21h

mov dl,al

int 21h

mov cx,100

h1:

mov ah,2

int 21h

Loop h1

mov ah,4ch

int 21h

main endp

end main

**OUTPUT:**

