VARSHA

22570063

MICROPROCESSOR PRACTICALS

Ques 1. Write a program to print ‘Hello World’.

.MODEL SMALL

.DATA

MSG1 DB"HELLO WORLD $"

.CODE

.STARTUP

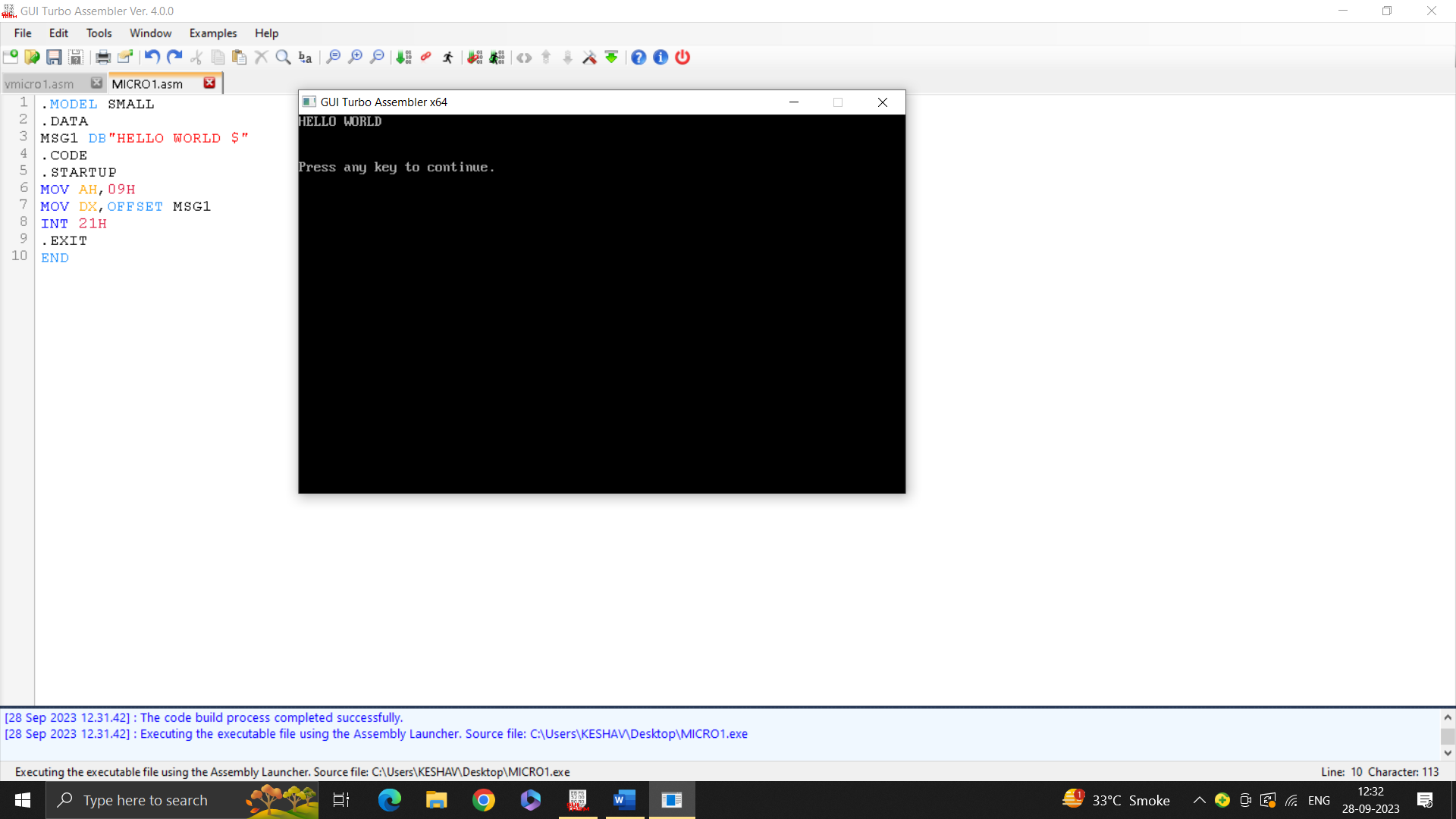
MOV AH,09H

MOV DX,OFFSET MSG1

INT 21H

.EXIT

END



Ques 2. Write a program to print two strings on two different lines.

.MODEL SMALL

.DATA

MSG1 DB"HELLO 4"

MSG2 DB 10,13, "WORLD $"

.CODE

.STARTUP

MOV AH,09H

MOV DX,OFFSET MSG1

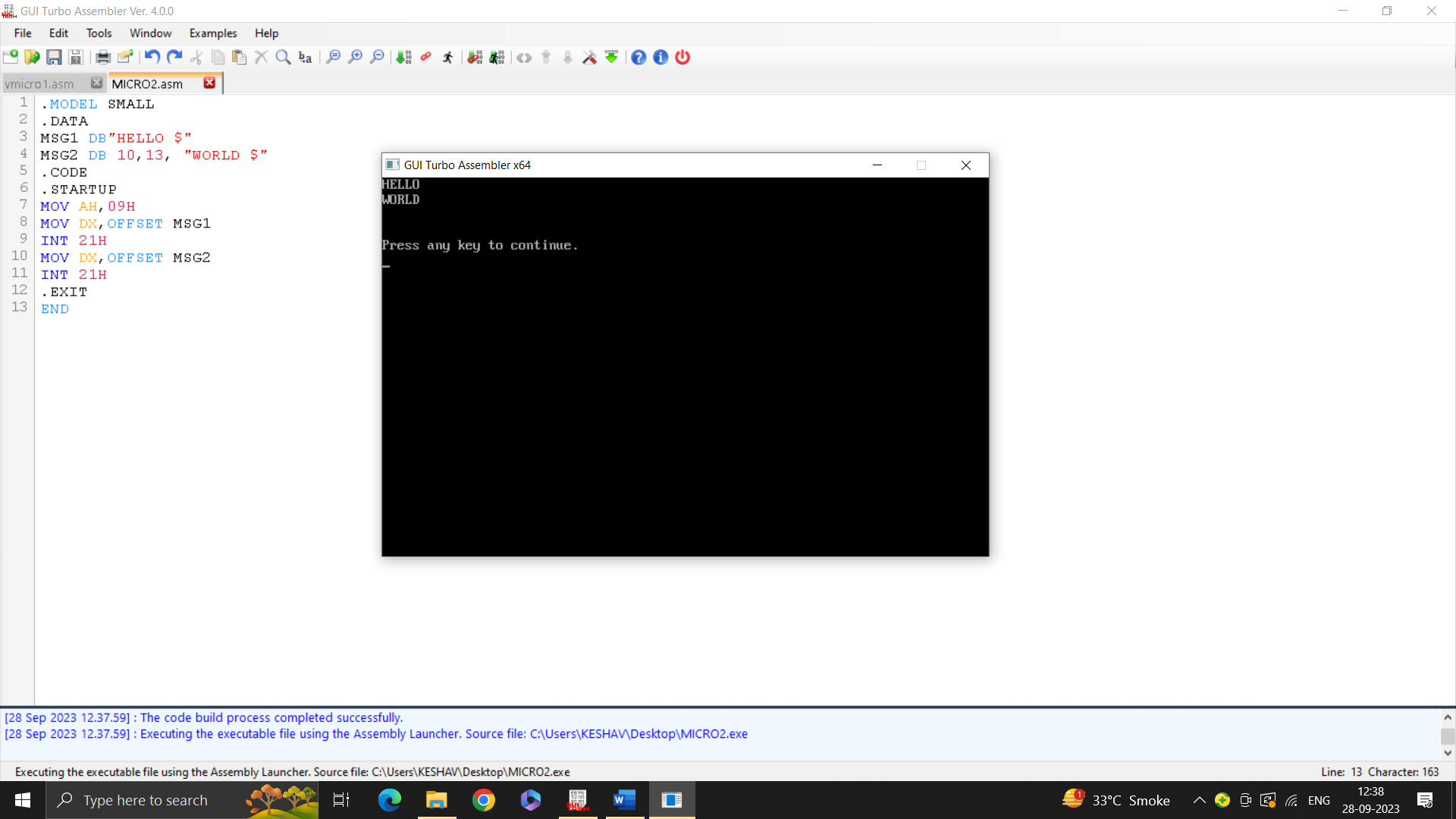
INT 21H

MOV DX,OFFSET MSG2

INT 21H

.EXIT

END



Ques 3. Write a program to take a single digit number from the user and print that number on the console.

.MODEL SMALL

.DATA

MSG1 DB"ENTER NUMBER: $"

MSG2 DB 10 , "ENTERED NUMBER IS: $"

.CODE

.STARTUP

MOV AH,09H

MOV DX,OFFSET MSG1

INT 21H

MOV AH,01H

INT 21H

MOV AH,09H

MOV DX,OFFSET MSG2

INT 21H

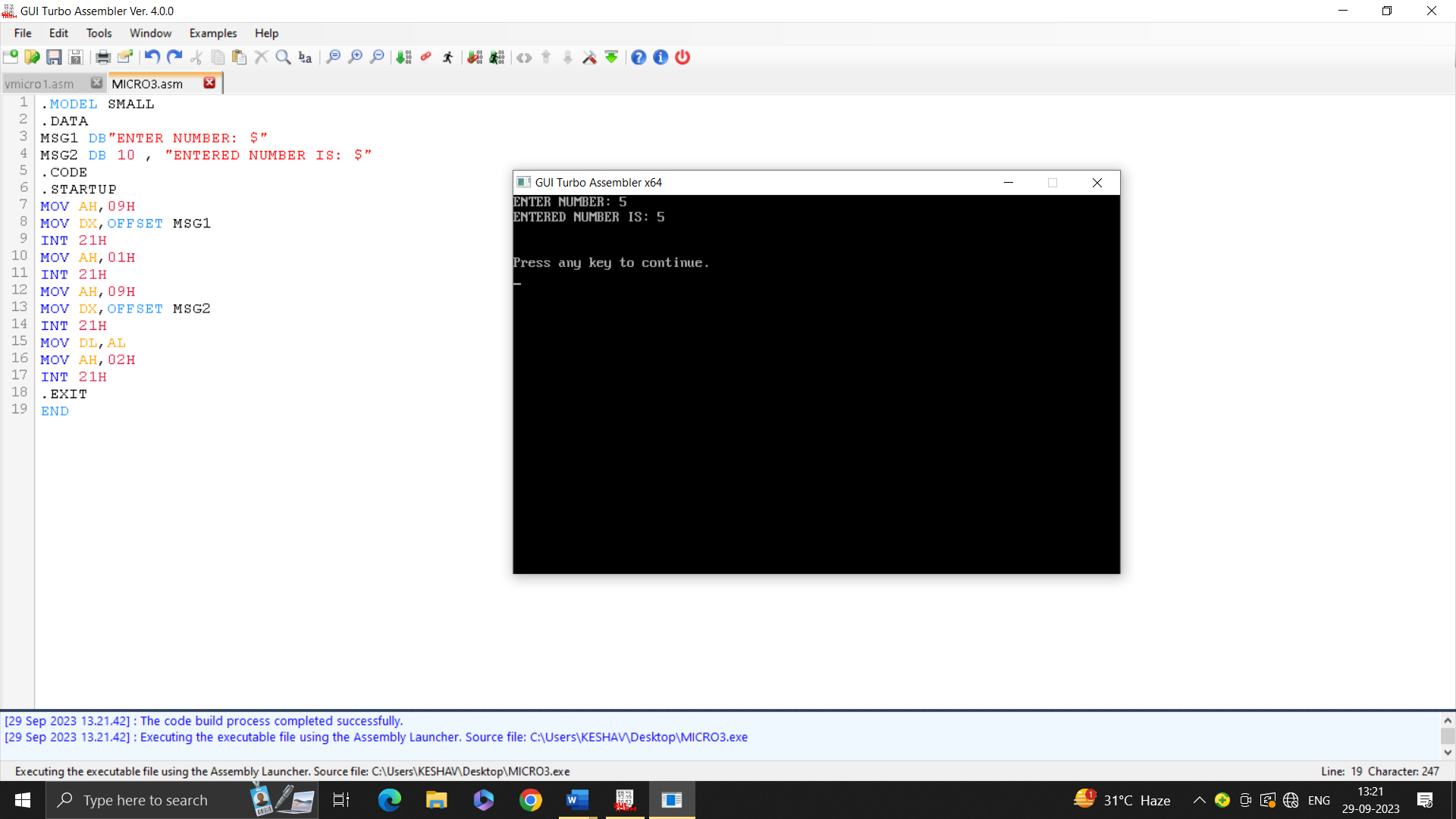
MOV DL,AL

MOV AH,02H

INT 21H

.EXIT

END



Ques 4. Write a program to compare two single digit numbers and check if they are equal or not.

.MODEL SMALL

.DATA

MSG1 db "Enter 1st Number: $"

MSG2 db 10,13,"Enter 2nd Number: $"

MSGEQUAL db 10 ,13, "NUMBERS ARE EQUAL $"

MSGNEQUAL db 10 ,13, "NUMBERS ARE NOT EQUAL !! $"

.CODE

.STARTUP

MOV AH,09H

MOV DX,OFFSET MSG1

INT 21H

MOV AH,01H

INT 21H

MOV BH,AL

MOV AH,09H

MOV DX,OFFSET MSG2

INT 21H

MOV AH,01H

INT 21H

MOV BH,AL

CMP BH,BL

JZ EQUAL

JNZ NOTEQUAL

EQUAL:

MOV AH,09H

MOV DX,OFFSET MSGEQUAL

INT 21H

MOV AH,4CH

INT 21H

NOTEQUAL:

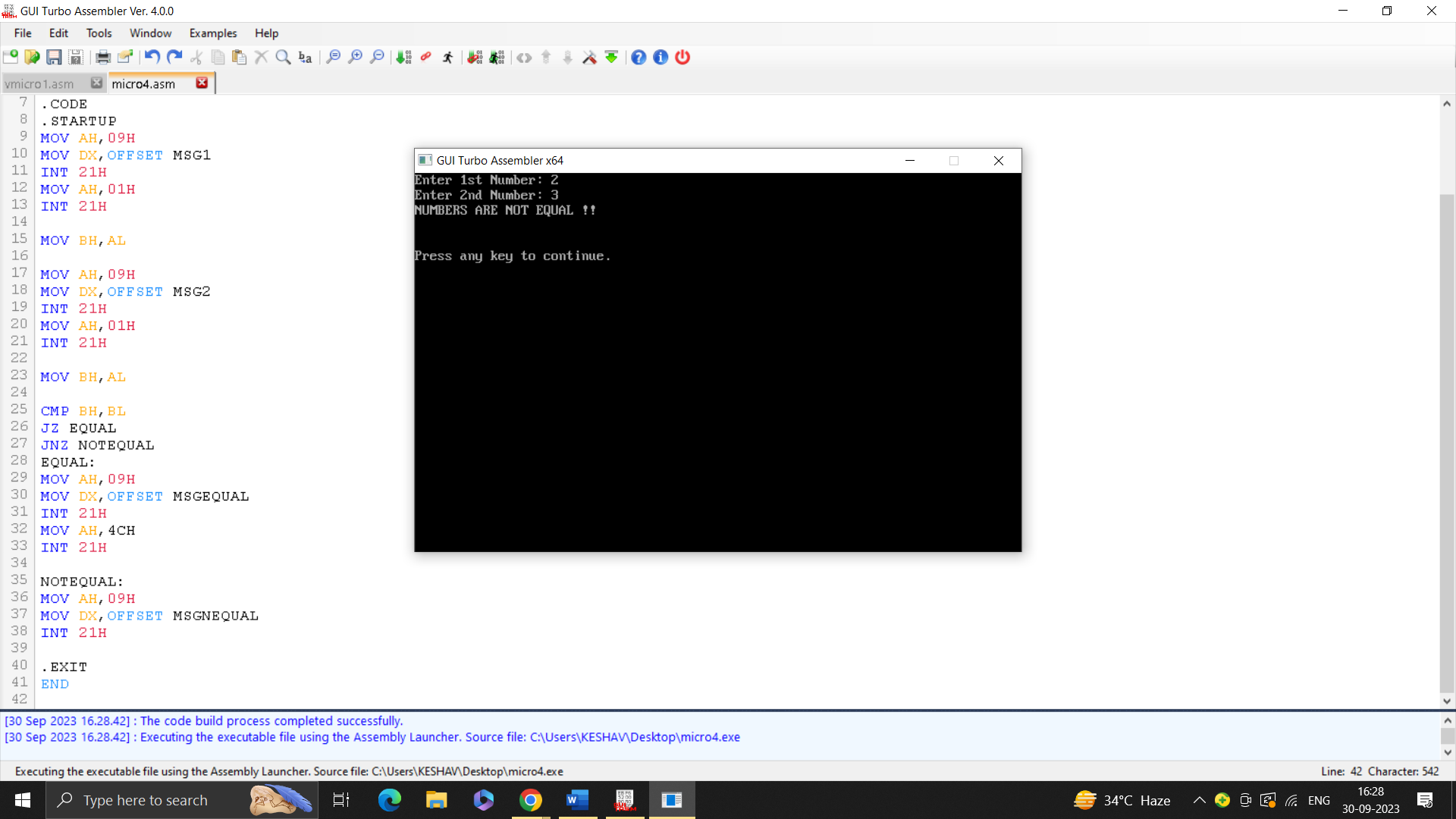
MOV AH,09H

MOV DX,OFFSET MSGNEQUAL

INT 21H

.EXIT

END



Ques 5. Write a program for 8-bit addition of two single digit numbers. Show the result after ASCII adjust.

.model small

.data

num1 db '5'

num2 db '3'

result db 0

.code

main proc

mov ax, @data

mov ds, ax

mov al, num1

mov bl, num2

sub al, 30h

sub bl, 30h

add al, bl

daa

mov result, al

mov ah, 02h

mov dl, result

int 21h

mov ah, 4Ch

int 21h

main endp

end main

Ques 6. Write a program for 16-bit addition of two double digit numbers. Show the result after ASCII adjust.

.model small

.data

num1 db '25'

num2 db '37'

result db 0

.code

main proc

mov ax, @data

mov ds, ax

mov ax, word ptr num1

mov bx, word ptr num2

sub ax, 3030h

sub bx, 3030h

add ax, bx

aaa

aas

mov result, al

mov ah, 02h

mov dl, al

int 21h

mov dl, ah

int 21h

mov ah, 4Ch

int 21h

main endp

end main

Ques 7. Write a program for 16-bit BCD addition.

.model small

.data

.code

main proc

mov ax, @data

mov ds, ax

mov ah, 12h

mov al, 34h

mov bh, 56h

mov bl, 78h

add al, bl

daa

adc ah, bh

daa

mov dl, ah

call display\_digit

mov dl, al

call display\_digit

mov ah, 4Ch

int 21h

main endp

display\_digit proc

push ax

push bx

push cx

mov bx, 10

xor cx, cx

display\_loop:

xor dx, dx

div bx

add dl, '0'

push dx

inc cx

test ax, ax

jnz display\_loop

display\_loop\_reverse:

pop dx

mov ah, 2

int 21h

loop display\_loop\_reverse

pop cx

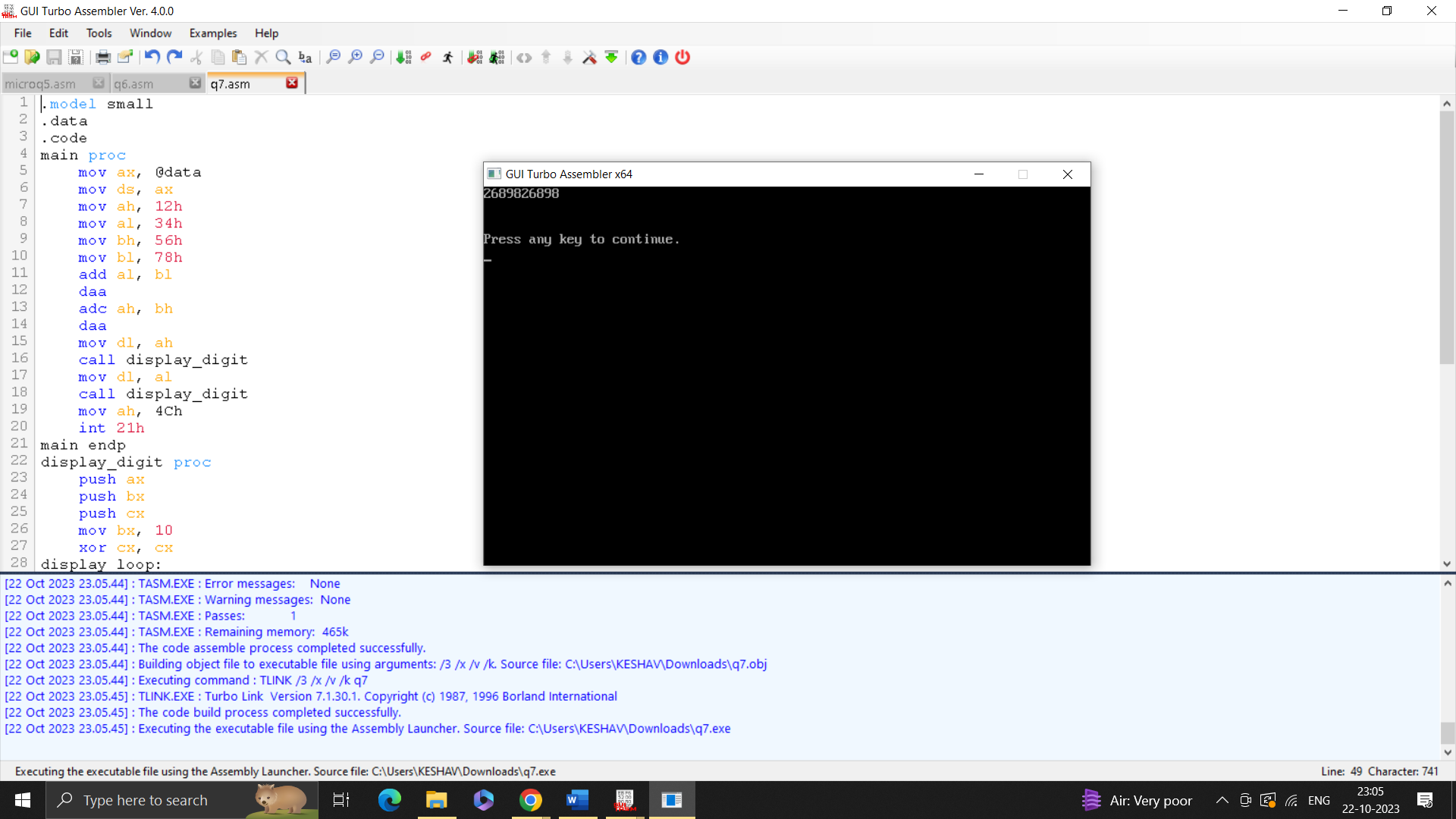
pop bx

pop ax

ret

display\_digit endp

end main



Ques 8. Write a program for 32-bit BCD addition and substraction.

.MODEL TINY

.CODE

.STARTUP

MOV AX,3214H

MOV BX,6212H

MOV CX,4321H

MOV DX,6543H

SUB AX,BX

SBB DX,CX

DAA

MOV BX,AX

MOV AX,DX

CALL DISPH

MOV AX,BX

CALL DISPH

.EXIT

DISPH PROC NEAR

MOV CL,4

MOV CH,4

DISPH1:

ROL AX,CL

PUSH AX

AND AL,0FH

ADD AL,30H

CMP AL,'9'

JBE DISPH2

ADD AL,7

DISPH2:

MOV AH,2

MOV DL,AL

INT 21H

POP AX

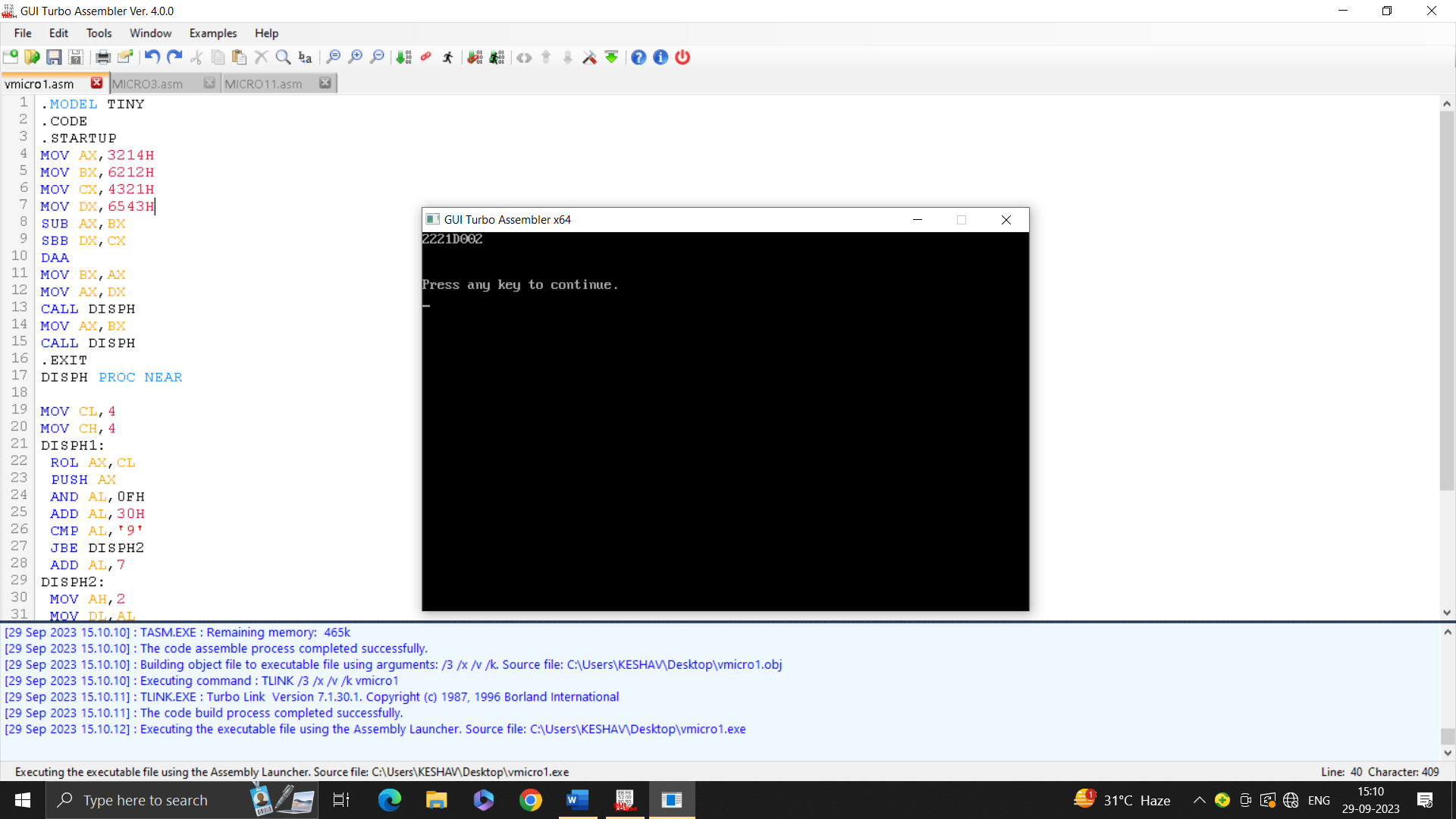
DEC CH

JNZ DISPH1

RET

DISPH ENDP

END



Ques9. Write a program for 32 bit binary addition, substraction,multiplication and division.

.MODEL TINY

.CODE

.STARTUP

MOV AX,3214H

MOV BX,6212H

MOV CX,4321H

MOV DX,6543H

ADD AX,BX

ADC DX,CX

MOV BX,AX

MOV AX,DX

CALL DISPH

MOV AX,BX

CALL DISPH

.EXIT

DISPH PROC NEAR

MOV CL,4

MOV CH,4

DISPH1:

ROL AX,CL

PUSH AX

AND AL,0FH

ADD AL,30H

CMP AL,'9'

JBE DISPH2

ADD AL,7

DISPH2:

MOV AH,2

MOV DL,AL

INT 21H

POP AX

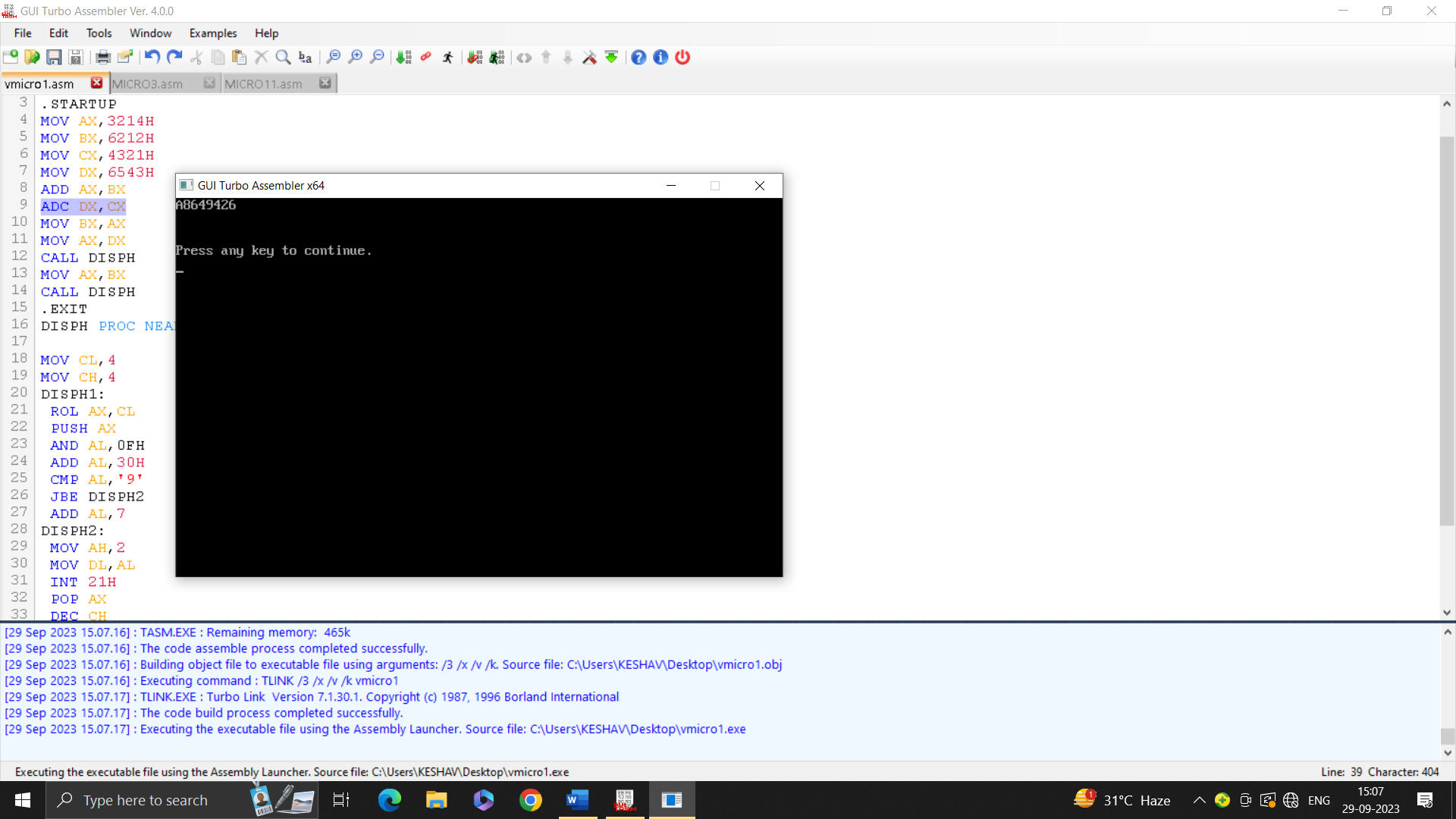
DEC CH

JNZ DISPH1

RET

DISPH ENDP

END



Ques 10.write a program for binary to ASCII conversion.

.MODEL SMALL

.data

MSG1 DB"EnTer binary no.: $"

MSG2 DB 10 , " ASCII: $"

.CODE

.STARTUP

MOV AH,09H

MOV DX,OFFSET MSG1

INT 21H

MOV BL,0

MOV CL,8

loops:

MOV AH,01H

INT 21H

SUB AL,48

SHL BL,1

ADD BL,AL

looP loops

MOV AH,09H

MOV DX,OFFSET MSG2

INT 21H

MOV AH,02H

MOV DL,BL

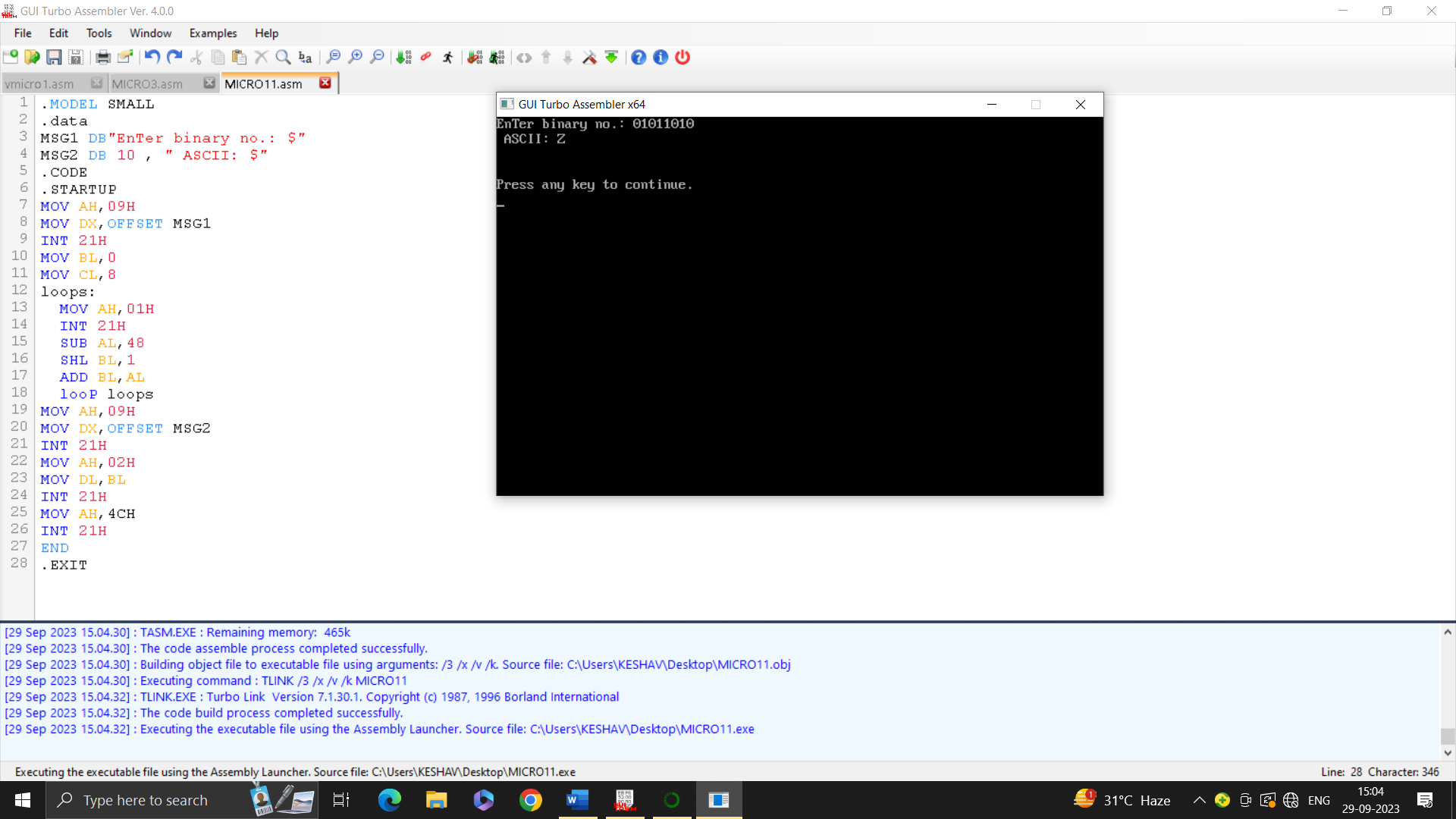
INT 21H

MOV AH,4CH

INT 21H

END

.EXIT



Ques 11. Write a program for ASCII to binary conversion.

.MODEL SMALL

.DATA

MSG1 DB"ENTER A NUMBER: $"

MSG2 DB 10 , "BINARY NUMBER IS: $"

.CODE

.STARTUP

MOV AH,09H

MOV DX,OFFSET MSG1

INT 21H

MOV BL,0

MOV AH,01H

INT 21H

MOV BL,AL

MOV AH,09H

MOV DX,OFFSET MSG2

INT 21H

MOV CL,8

loops:

MOV AH,02H

SHL BL,1

JC printtone

MOV DL,48

JMP Justprint

Printtone:

MOV DL,49

Justprint:

INT 21H

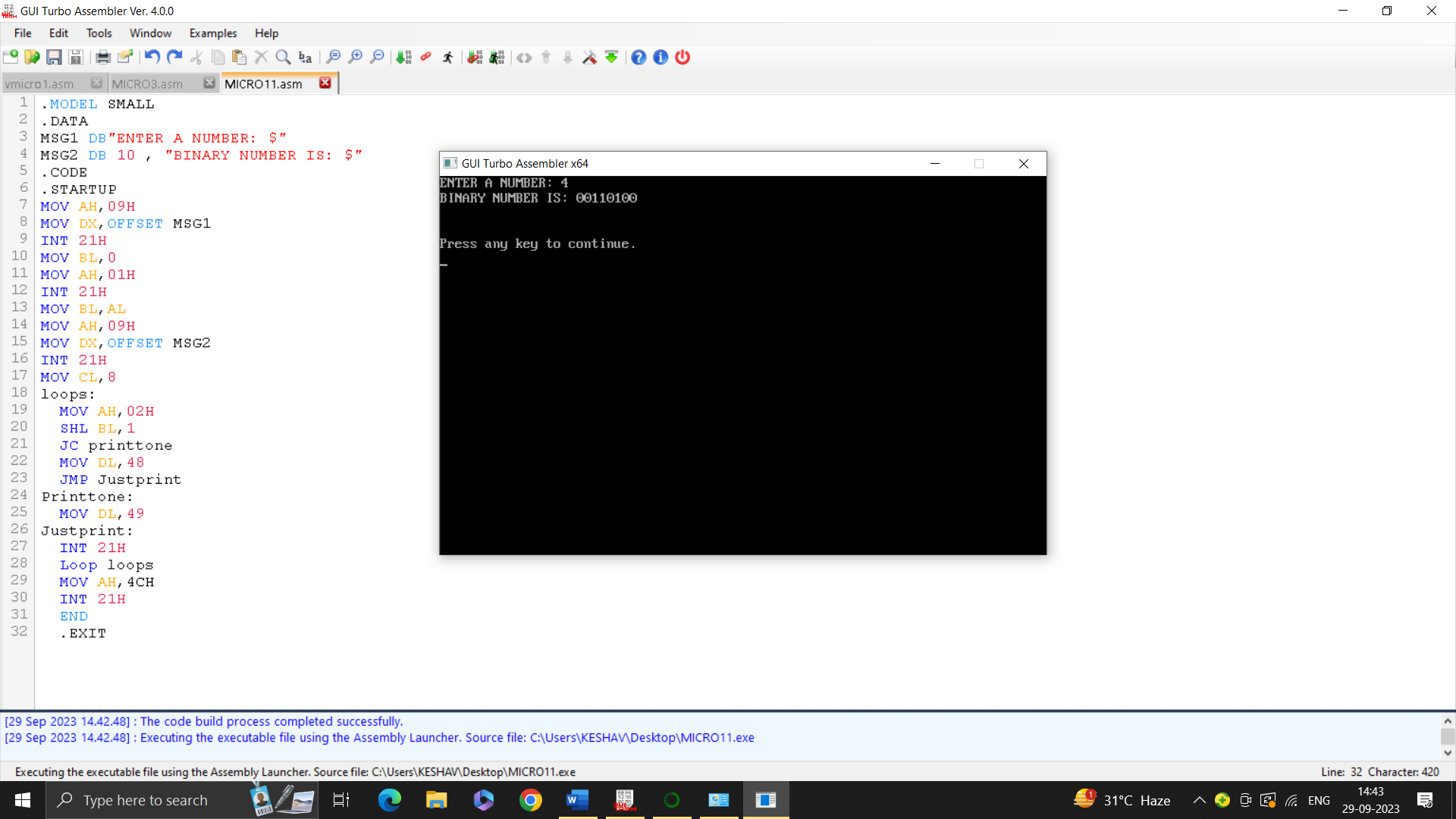
Loop loops

MOV AH,4CH

INT 21H

END

.EXIT



Ques 12. Write a program to take input in an array and print it on the console.

.MODEL SMALL

.DATA

MSG1 DB "ENTER SIZE OF AN ARRAY$"

MSG2 DB 10,"ENTER THE ELEMENTS OF ARRAY$"

MSG3 DB 10,"ARRAY ELEMENTS ARE:$"

ARR DB 20 DUP(?)

N DB ?

.CODE

.STARTUP

MOV AH,09H

MOV DX, OFFSET MSG1

INT 21H

MOV AH,01H

INT 21H

SUB AL,30H

MOV N,AL

MOV SI,0

INPUT:

MV AH,09H

MOV DX,OFFSET MSG2

INT 21H

MOV AH,01H

INT 21H

MOV ARR[SI],AL

INC SI

DEC N

JNZ INPUT

OUTPUT:

MOV CH,0

MOV CX,SI

MOV SI,0

MOV AH,09H

MOV DX,OFFSET MSG3

INT 21H

OUTPUT\_LOOP:

MOV DL,ARR[SI]

MOV AH,02H

INT 21H

MOV DL,20H

INT 21H

INC SI

LOOP OUTPUT\_LOOP

.EXIT

END

Ques 13. Write a program to sort an array using bubble sort.

.model small

.data

N1 db ?

MSG1 db 10,"Enter the size of array:$"

MSG2 db 10,"Enter array elements:",10,"$"

MSG3 db 10,"Entered values are:",10,"$"

MSG4 db 10,"Array after Bubble sort :",10,"$"

ARR db 50 DUP(?)

.code

.startup

MOV DX,offset MSG1

MOV AH,09H

INT 21H

MOV DI,offset N1

CALL takeInput

MOV DX,offset MSG2

MOV AH,09H

INT 21H

MOV DI,offset ARR

MOV CL,N1

MOV CH,0

L1: CALL takeInput

INC DI

LOOP L1

MOV DX,offset MSG3

MOV AH,09H

INT 21H

MOV DI,offset ARR

MOV CL,N1

MOV CH,0

L2: CALL PRINT

MOV DL,' '

MOV AH,02H

INT 21H

INC DI

LOOP L2

CALL SORT

MOV DX,offset MSG4

MOV AH,09H

INT 21H

MOV DI,offset ARR

MOV CL,N1

MOV CH,0

L3: CALL PRINT

MOV DL,' '

MOV AH,02H

INT 21H

INC DI

LOOP L3

.exit

takeInput PROC NEAR

MOV BX,0H

ScanNum:

MOV AH,01H

INT 21H

CMP AL,13

JE EXIT

SUB AL,30H

MOV AH,0H

PUSH AX

MOV AL,BL

MOV DL,10

MUL DL

POP DX

ADD AX,DX

MOV BX,AX

JMP ScanNum

EXIT:

MOV[DI],BX

RET

takeInput ENDP

PRINT PROC NEAR

MOV AL,[DI]

MOV AH,0

MOV DL,10

DIV DL

PUSH AX

MOV DL,AL

ADD DL,30H

MOV AH,02H

INT 21H

POP AX

MOV DL,AH

ADD DL,30H

MOV AH,02H

INT 21H

RET

PRINT ENDP

SORT PROC NEAR

MOV CL,N1

MOV CH,0

DEC CX

Outer: PUSH CX

MOV CL,N1

MOV CH,0

DEC CX

Inner: MOV BX,CX

MOV AL,ARR[BX]

DEC BX

MOV DL,ARR[BX]

CMP AL,DL

JA noOp

XCHG AL,DL

MOV ARR[BX],DL

INC BX

MOV ARR[BX],AL

noOp:

LOOP Inner

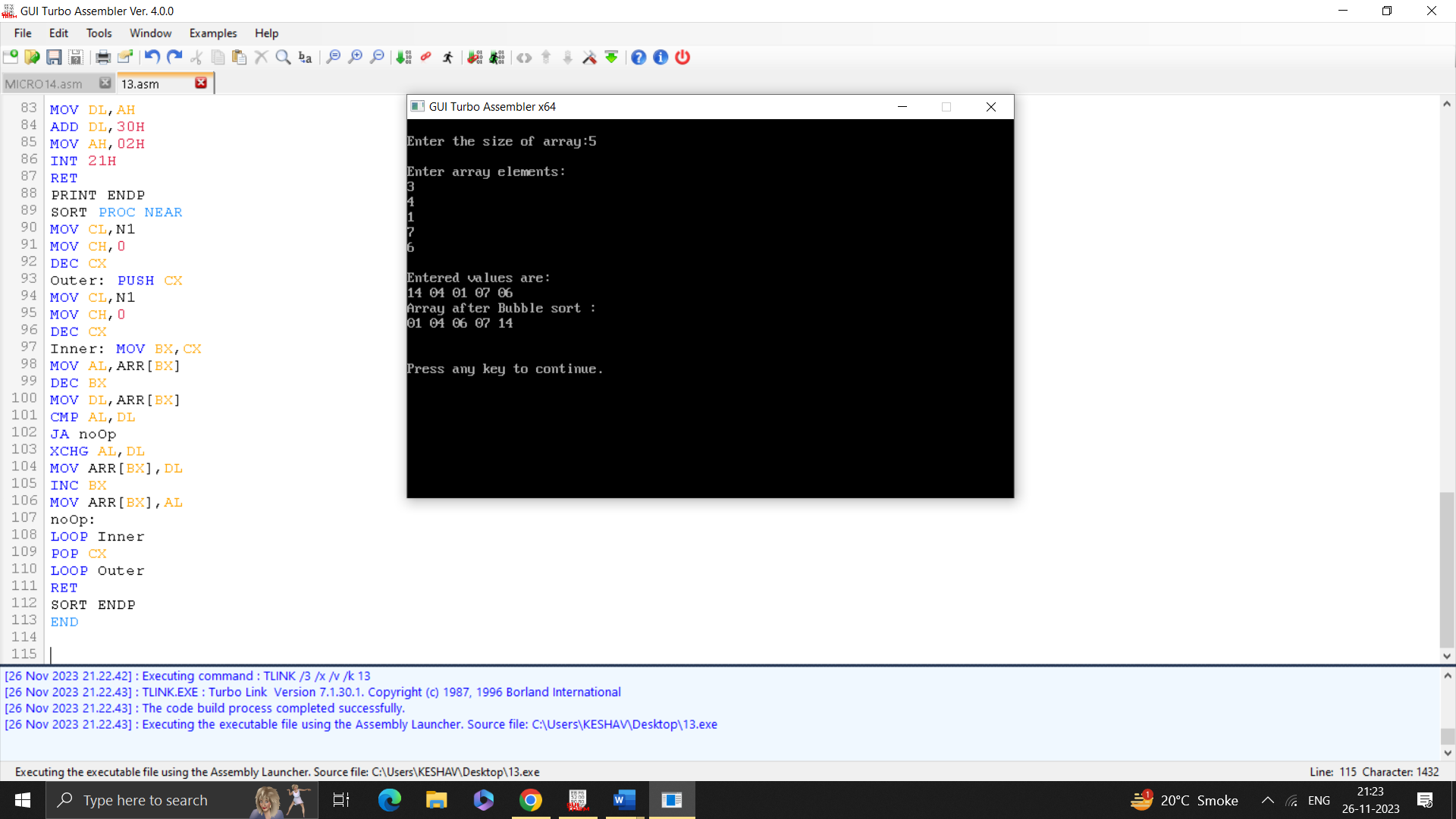
POP CX

LOOP Outer

RET

SORT ENDP

END



Ques 14. Write a program to perform linear search in an array.

.model small

.data

array db 8, 4, 10, 2, 6, 3, 7, 1

array\_size equ 8

search\_value db 6

found\_msg db "Value found at index: $"

not\_found\_msg db "Value not found.$"

result dw 0

.code

main proc

mov ax, @data

mov ds, ax

lea si, array

mov al, search\_value

mov cx, array\_size

search\_loop:

cmp al, [si]

je found

inc si

loop search\_loop

not\_found:

lea dx, not\_found\_msg

mov ah, 09h

int 21h

jmp done

found:

mov result, si

lea dx, found\_msg

mov ah, 09h

int 21h

mov ax, result

mov cx, 10

xor dx, dx

div cx

push dx

add dl, '0'

mov ah, 02h

int 21h

pop dx

add dl, '0'

mov ah, 02h

int 21h

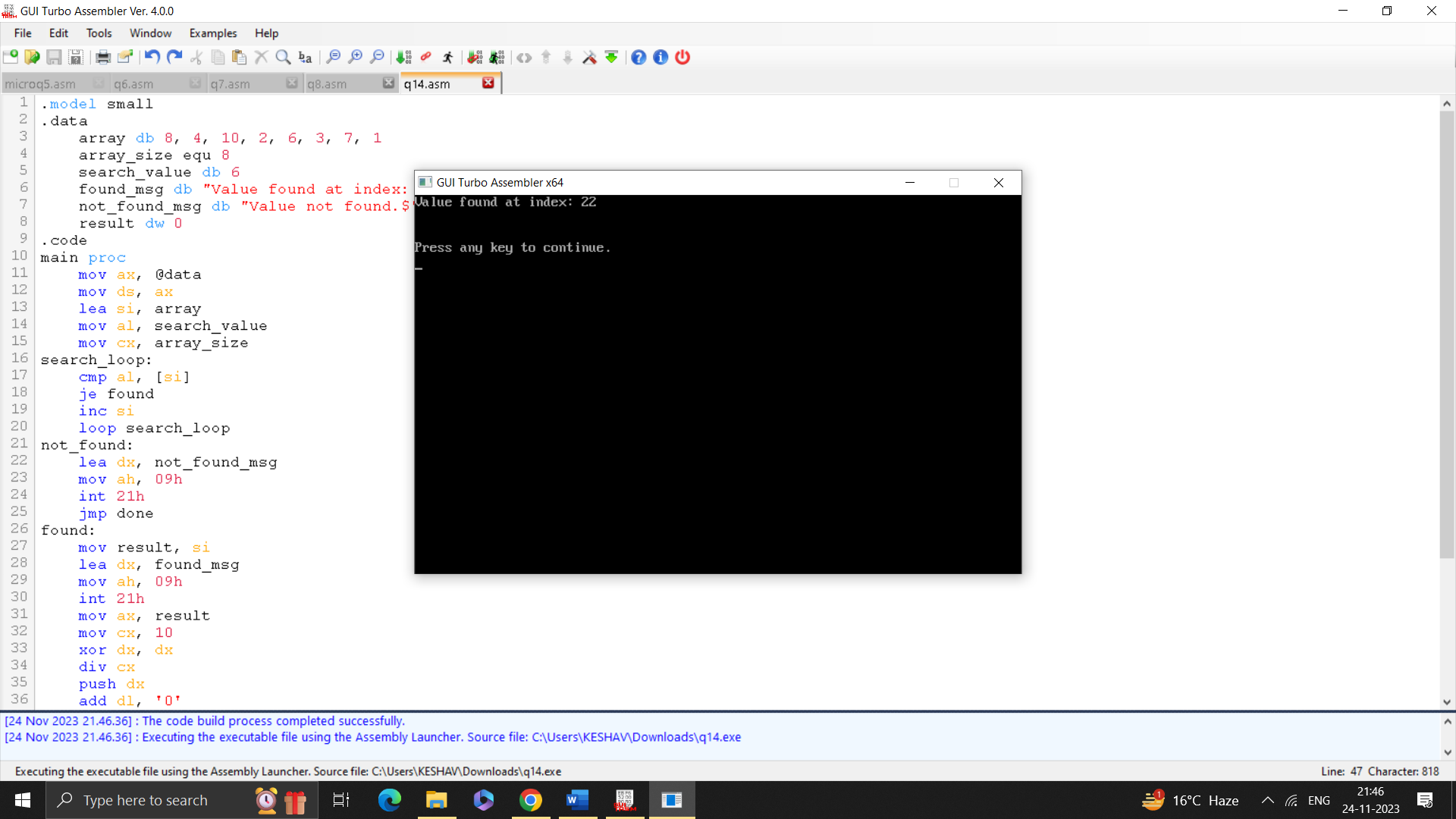
done:

mov ah, 4Ch

int 21h

main endp

end main



Ques 15. Write a program to perform binary search in an array.

.MODEL SMALL

.DATA

MSG1 DB "ENTER ARRAY SIZE : $"

MSG2 DB 10,"ENTER ELEMENT OF ARRAY(SORTED ARRAY): $"

MSG3 DB 10,"ENTER ELEMENT TO BE SEARCHED: $"

MSG4 DB 10,"ELEMENT FOUND AT LOCATION$"

MSG5 DB 10,"ELEMENT NOT FOUND$"

ARR DB 20 DUP(?)

N DB ?

P DB ?

COUNT DW ?

.CODE

.STARTUP

MOV AH,09H

MOV DX,OFFSET MSG1

INT 21H

MOV AH,01H

INT 21H

SUB AL,30H

MOV N,AL

MOV CH,0

MOV SI,0

MOV CL,N

MOV COUNT,CX

INPUT:

MOV AH,09H

MOV DX,OFFSET MSG2

INT 21H

MOV AH,01H

INT 21H

MOV ARR[SI],AL

INC SI

LOOP INPUT

MOV AH,09H

MOV DX,OFFSET MSG3

INT 21H

MOV AH,01H

INT 21H

MOV P,AL

MOV BX,OFFSET ARR

MOV CH,0

MOV CL,N

MOV DI,0

MOV AH,0

MOV AL,0

DEC SI

OUTPUT:

DEC COUNT

ADD AX, CX

MOV DL, 2

DIV AL

MOV AH,0

MOV DI,AX

MOV AL,BX[DI]

CMP COUNT,SI

JG NF

CMP P,AL

JG GREATER

CMP P ,AL

JL LESS

CMP P,AL

JE FOUND

NF:

MOV AH,09H

MOV DX,OFFSET MSG5

INT 21H

.EXIT

GREATER:

MOV AX,DI

JMP OUTPUT

LESS:

MOV AH,0

MOV AL,0

MOV CX,DI

JMP OUTPUT

FOUND:

MOV AH,09H

MOV DX,OFFSET MSG4

INT 21H

MOV DX,DI

INC DX

MOV AX,DX

CALL DISPH

.EXIT

DISPH PROC NEAR

MOV CL, 4

MOV CH, 4

DISPH1:

ROL AX,CL

PUSH AX

AND AL,0FH

ADD AL,30H

CMP AL, '9'

JBE DISPH2

ADD AL, 7

DISPH2:

MOV AH,2

MOV DL,AL

INT 21H

POP AX

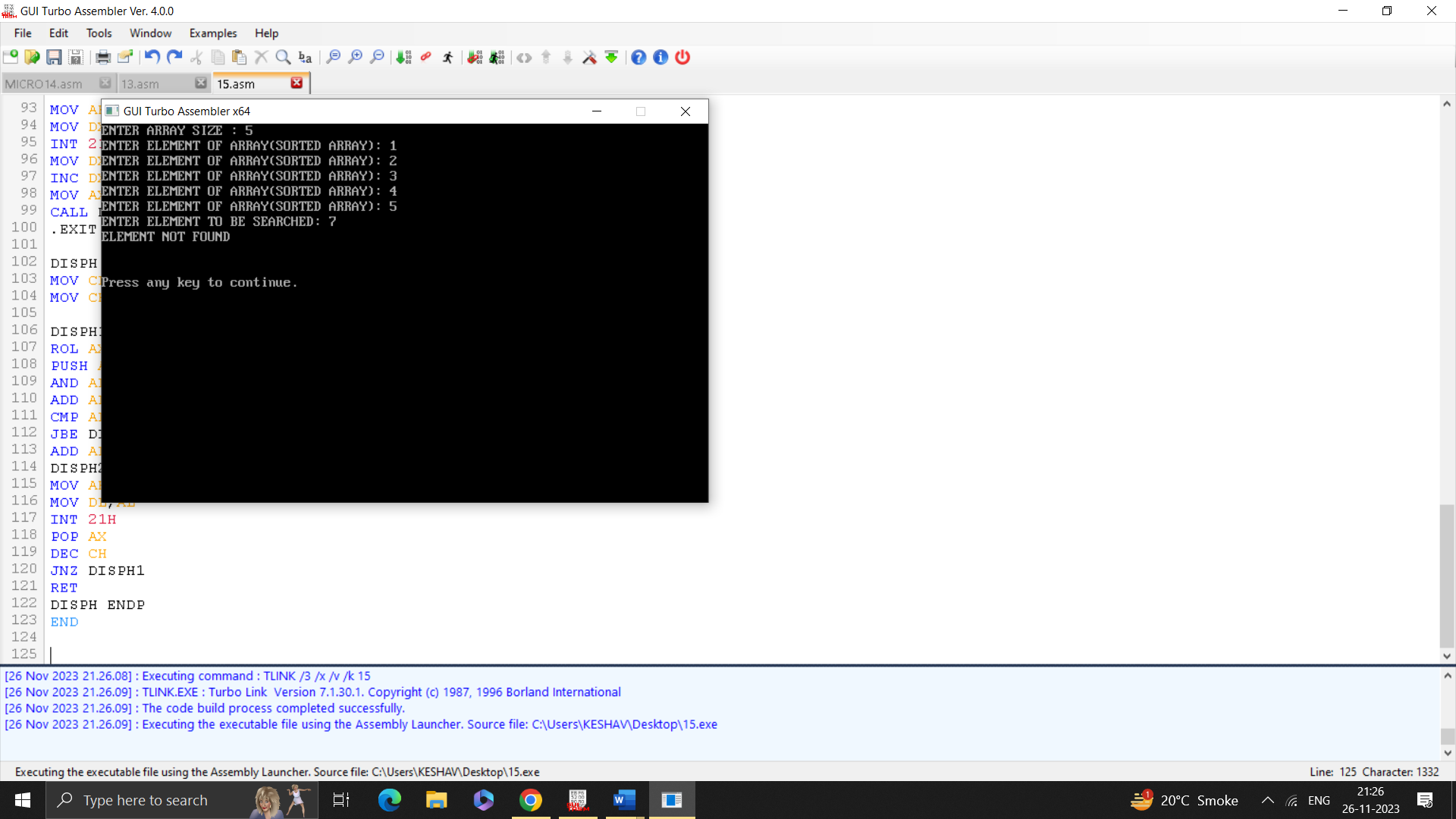
DEC CH

JNZ DISPH1

RET

DISPH ENDP

END



Ques 16.Write a program to program to add and substract two arrays.

.MODEL SMALL

.DATA

array1 DB 5, 4, 5, 2, 7

array2 DB 2, 1, 3, 1, 2

result\_add DB 11 DUP(?), '$'

result\_sub DB 11 DUP(?), '$'

.CODE

\_start:

MOV AX, @DATA

MOV DS, AX

*; Add arrays*

MOV SI, offset array1

MOV DI, offset array2

MOV BX, offset result\_add

MOV CX, 5

ADD\_LOOP:

MOV AL, [SI]

ADD AL, [DI]

ADD AL, '0' *; Convert to ASCII*

MOV [BX], AL

INC SI

INC DI

INC BX

LOOP ADD\_LOOP

*; Print the addition result*

MOV AH, 09h *; DOS print string function*

MOV DX, offset result\_add

INT 21h

*; Subtract arrays*

MOV SI, offset array1

MOV DI, offset array2

MOV BX, offset result\_sub

MOV CX, 5

SUB\_LOOP:

MOV AL, [SI]

SUB AL, [DI]

ADD AL, '0' *; Convert to ASCII*

MOV [BX], AL

INC SI

INC DI

INC BX

LOOP SUB\_LOOP

*; Print the subtraction result*

MOV AH, 09h *; DOS print string function*

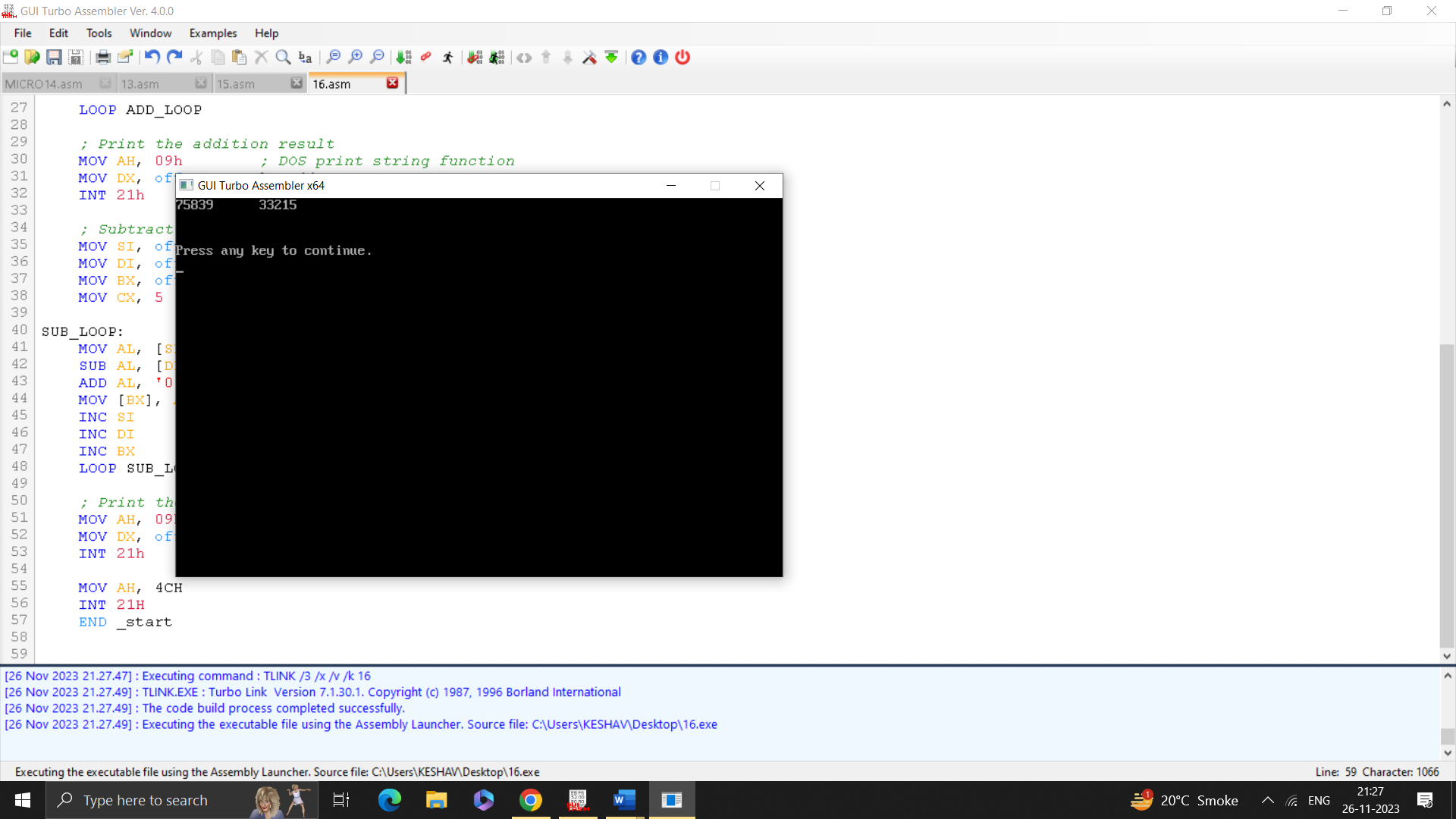
MOV DX, offset result\_sub

INT 21h

MOV AH, 4CH

INT 21H

END \_start



Ques 17. Write a program to interface a microprocessor with external devices such as keyboard and elevator.