COA – Lab Assignment 3

Name: Bhavin Patil

Roll No- 78

GR No : 12120056

Div : CS- D

Batch : B3

Problem Statement : Accept 5 subjects marks of student, find its average and display the grade scored by student: Make use of array initialization, DIV instruction, CALL instruction and near procedure

Instructions –

- MOV: This instruction is used to move data from one location to another.
  Syntax – mov destination, source
- LEA (Load Effective Address): It loads the specified register with the offset of a memory location.
- ADD: it performs an addition on both the first source register's contents and the second source. register's contents, and stores the result in the destination register
- DIV: The DIV (unsigned divide) instruction performs 8-bit, 16-bit, and 32-bit division on unsigned integers.
- CALL: CALL instruction is used to call a subroutine. Subroutines are often used to perform tasks that need to be performed frequently.

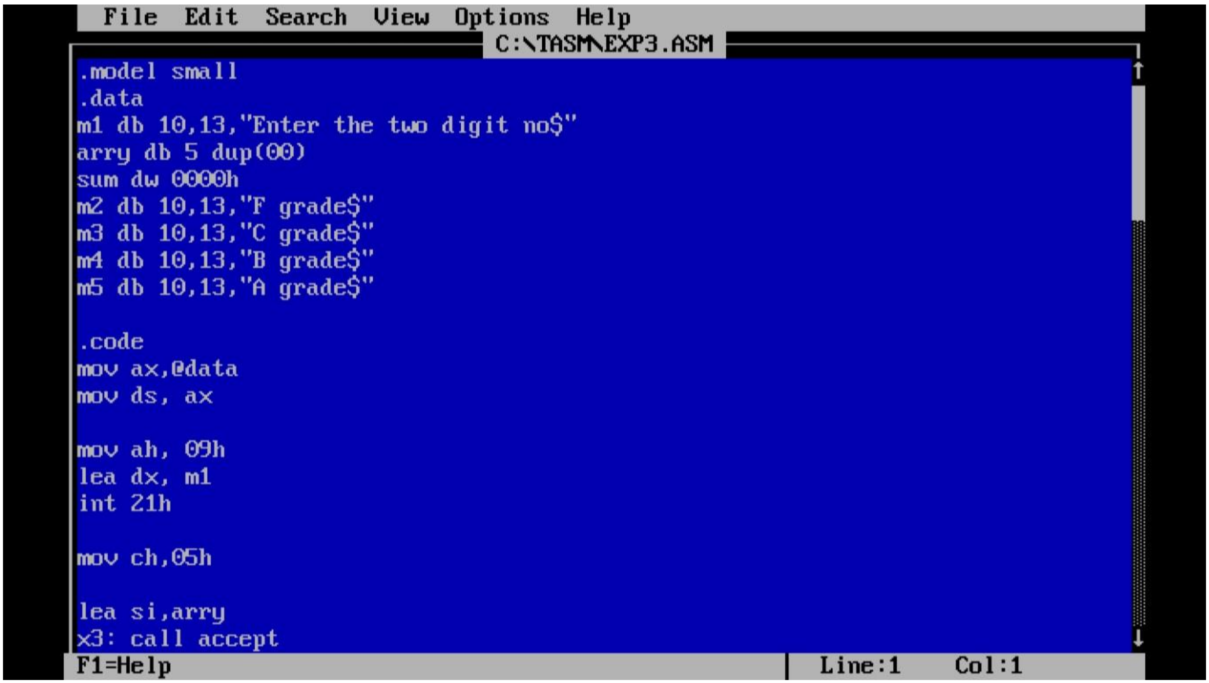
Commands –

1. **01h :** It is used to read character from standard input, with echo, result is stored in AL.
2. **02h :** It is used to display single character
3. **09h**: Displays the string until “$” is reached.
4. **Int 21h**: Interrupt used to exit the program.

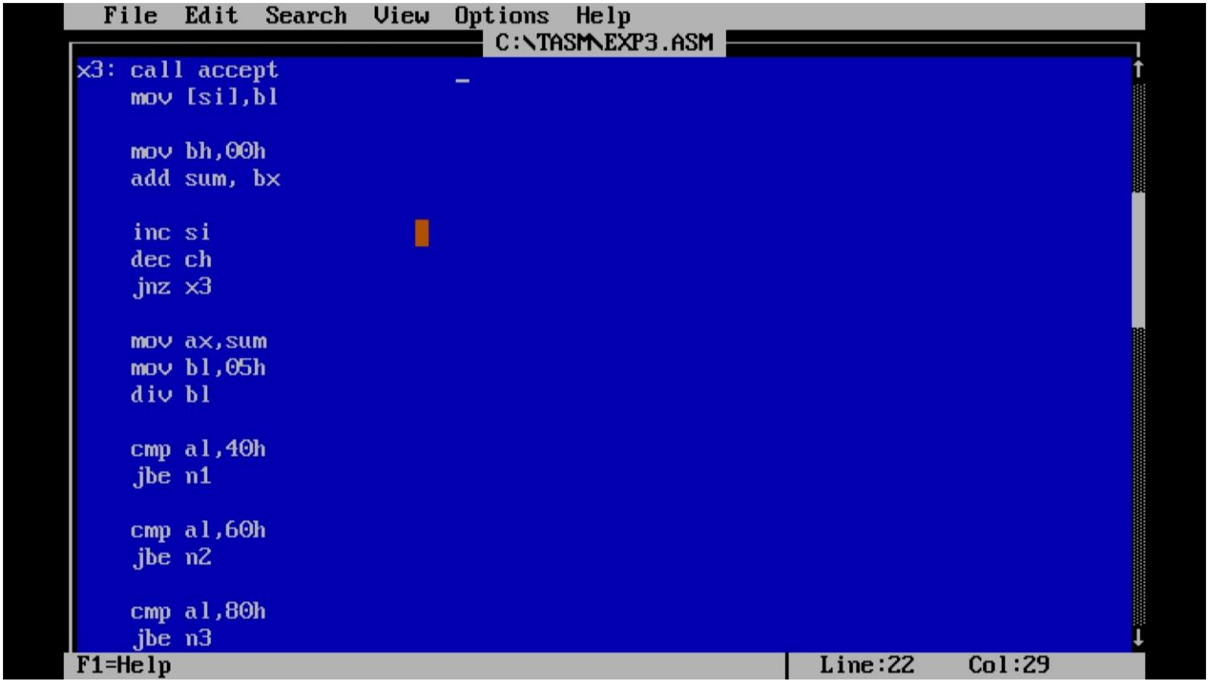
5. **.data**: This Command is used only when we want to store in Data Segment, basically, it is the memory access of the Data Segment. Whatever we want to print must be written here. Also, the variables are declared here.
6. **10, 13**: They work as Escape Sequence Character
7. **$**: It states the end of a Statement
8. **Db (Define Byte)**: It acts as an Assembler Directive
9. **. code**: Full Logical Program is written here
10. **Tasm –** Used for Compilation
11. **tlink**– Perform linking operation

Screenshots of Source Code and Output:

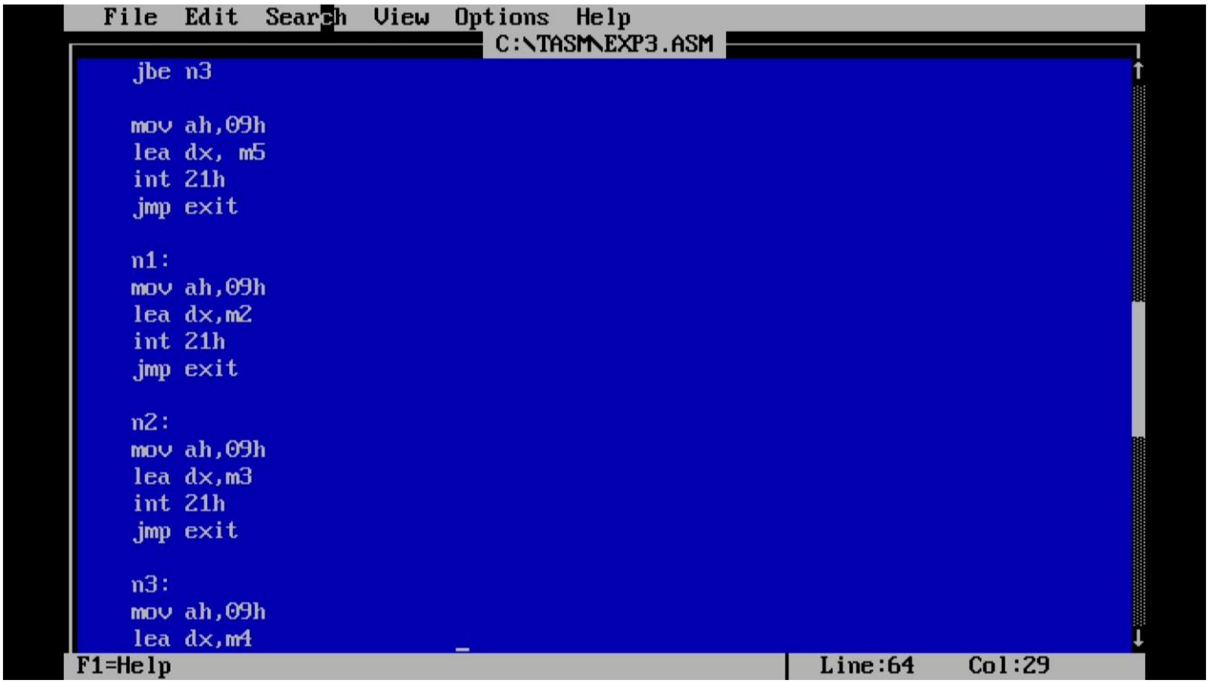
Source Code -



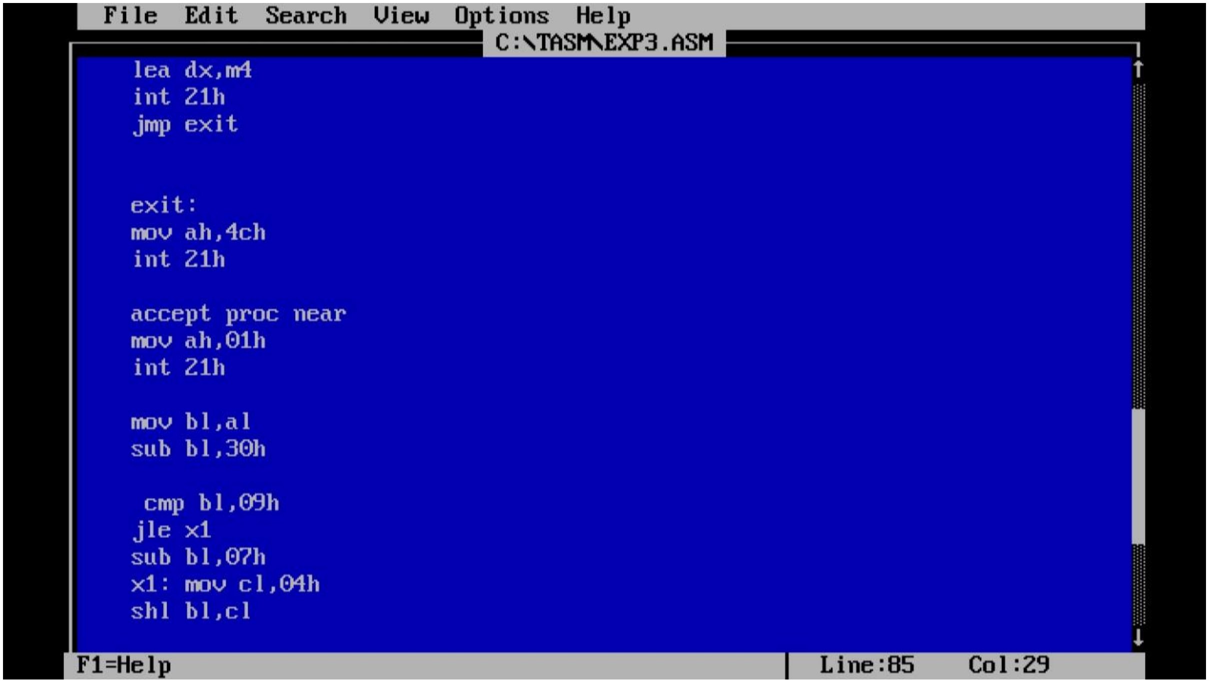
```
.model small
.data
m1 db 10,13,"Enter the two digit no$"
arry db 5 dup(00)
sum dw 0000h
m2 db 10,13,"F grade$"
m3 db 10,13,"C grade$"
m4 db 10,13,"B grade$"
m5 db 10,13,"A grade$"

.code
mov ax,@data
mov ds, ax

mov ah, 09h
lea dx, m1
int 21h

mov ch,05h

lea si,arry
x3: call accept
```



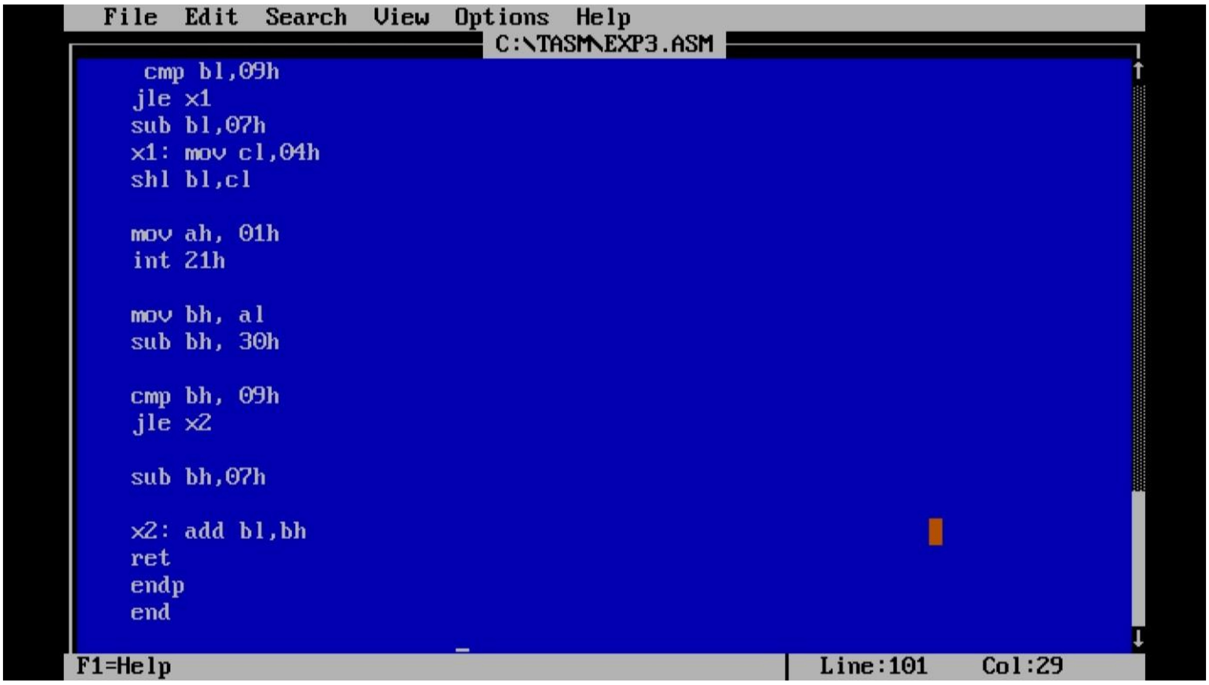
```
x3: call accept
    mov [si],bl

    mov bh,00h
    add sum, bx

    inc si
    dec ch
    jnz x3

    mov ax,sum
    mov bl,05h
    div bl

    cmp al,40h
    jbe n1

    cmp al,60h
    jbe n2

    cmp al,80h
    jbe n3
```



```
jbe n3

mov ah,09h
lea dx, m5
int 21h
jmp exit

n1:
mov ah,09h
lea dx,m2
int 21h
jmp exit

n2:
mov ah,09h
lea dx,m3
int 21h
jmp exit

n3:
mov ah,09h
lea dx,m4
```

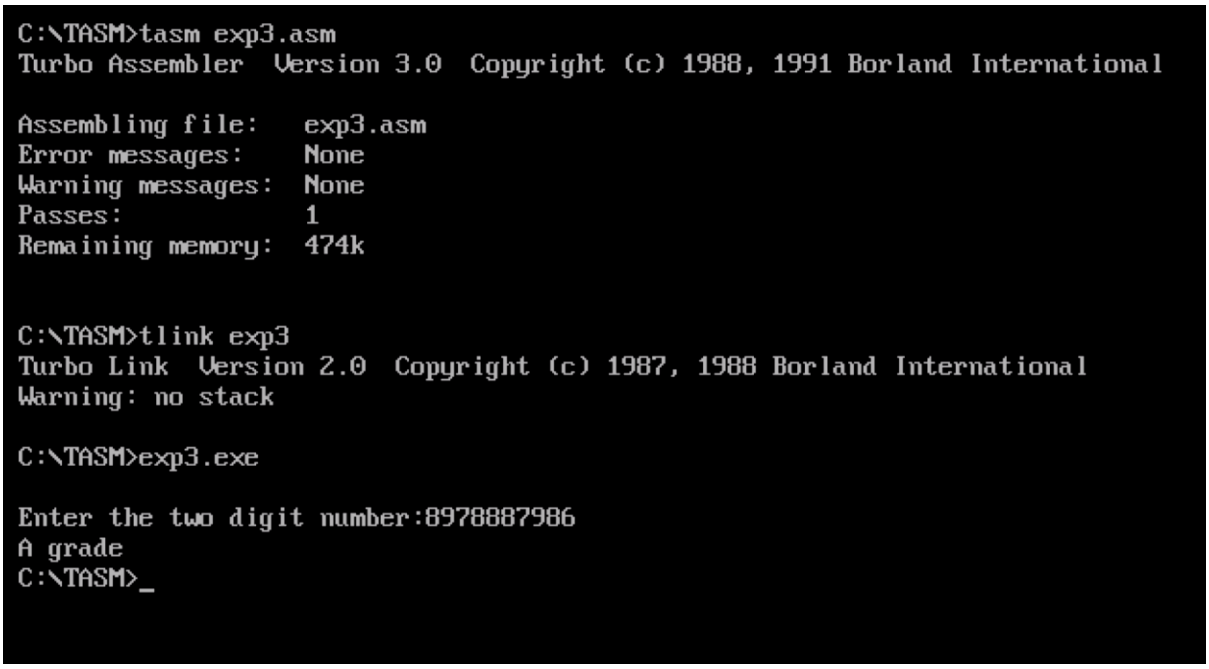


```
lea dx,m4
int 21h
jmp exit


exit:
mov ah,4ch
int 21h

accept proc near
mov ah,01h
int 21h

mov bl,al
sub bl,30h

 cmp bl,09h
jle x1
sub bl,07h
x1: mov cl,04h
shl bl,cl
```



```
 cmp bl,09h
jle x1
sub bl,07h
x1: mov cl,04h
shl bl,cl

mov ah, 01h
int 21h

mov bh, al
sub bh, 30h

cmp bh, 09h
jle x2

sub bh,07h

x2: add bl,bh
ret
endp
end
```

Output:



```
C:\TASM>tasm exp3.asm
Turbo Assembler  Version 3.0  Copyright (c) 1988, 1991 Borland International

Assembling file:   exp3.asm
Error messages:    None
Warning messages:  None
Passes:            1
Remaining memory:  474k


C:\TASM>tlink exp3
Turbo Link  Version 2.0  Copyright (c) 1987, 1988 Borland International
Warning: no stack

C:\TASM>exp3.exe

Enter the two digit number:8978887986
A grade
C:\TASM>_
```