NOTE: PRINT AR JONNO DL OR DX & INPUT AR JONNO AL OR AX , AFTER STRING USE $ SIGN.

SIMPLE INPUT & OUTPUT

```
.MODEL SMALL
.STACK 100H
.CODE

MAIN PROC
    MOV AH,1      ;INPUT
    INT 21H
    MOV BL,AL

    MOV AH,2      ;OUTPUT
    MOV DL,BL
    INT 21H

    MAIN ENDP
END MAIN
```

SIMPLE INPUT & OUTPUT WITH NEWLINE

```
.MODEL SMALL
.STACK 100H
.CODE

MAIN PROC
    MOV AH,1      ;INPUT
    INT 21H
    MOV BL,AL

    MOV AH,2     ; NEWLINE
    MOV DL,10
    INT 21H

    MOV AH,2    ;CRET
    MOV DL,13
    INT 21H


    MOV AH,2     ;OUTPUT
    MOV DL,BL
    INT 21H

    MAIN ENDP
END MAIN
```

INPUT & OUTPUT IN VARIABLE

```
.MODEL SMALL
.STACK 100H
.DATA
A DB 2
B DB ?

.CODE

MAIN PROC
    MOV AX,@DATA
    MOV DS,AX

    MOV AH,1       ;INPUT B
    INT 21H
    MOV B,AL

    MOV AH,2       ;NEWLINE
    MOV DL,10
    INT 21H
    MOV DL,13
```

```
   INT 21H

   MOV AH,2      ; PRINT A
   MOV DL,A
   ADD DL,48     ; CONVERT DECIMAL VALUE
   INT 21H

   MOV AH,2      ; PRINT B
   MOV DL,B
   INT 21H


   MAIN ENDP
END MAIN
```

SIMPLE STING OUTPUT

```
.MODEL SMALL
.STACK 100H
.DATA
A DW 'MOMIN $'

.CODE

MAIN PROC
   MOV AX,@DATA
   MOV DS,AX

   MOV AH,9   ; PRINT MOMIN
   LEA DX,A
   INT 21H


   MAIN ENDP
END MAIN
```

STRING INPUT & OUTPUT WITH FUNCTION

```
.MODEL SMALL
.STACK 100H
.DATA
A DW 'ENTER THE NUMBER : $'
B DW 'THE NUMBER IS : $'
NEWLINE DW 13,10,'$'  ;FOR NEWLINE

.CODE

NEWLINE MACRO  ; CREATE FUNCTION FOR NEWLINE
    MOV AH,2
    MOV DL,10
    INT 21H
    MOV DL,13
    INT 21H
ENDM

MAIN PROC
    MOV AX,@DATA
    MOV DS,AX

    MOV AH,9   ; PRINT A
    LEA DX,A
    INT 21H

    MOV AH,1     ;INPUT NUMBER
    INT 21H
    MOV BL,AL

    NEWLINE


    MOV AH,9       ;PRINT B
    LEA DX,B
    INT 21H

    MOV AH,2        ; PRINT NUMBER
    MOV DL,BL
    INT 21H



    MAIN ENDP
END MAIN
```

ADD TWO NUMBER

```
.MODEL SMALL
.STACK 100H
.DATA

.CODE

NEWLINE MACRO  ; CREATE FUNCTION FOR NEWLINE
    MOV AH,2
    MOV DL,10
    INT 21H
    MOV DL,13
    INT 21H
ENDM

MAIN PROC

    MOV AH,1    ;INPUT 1
    INT 21H
    MOV BL,AL

    NEWLINE
    MOV AH,1     ;INPUT 2
    INT 21H
    MOV BH,AL

    ADD BL,BH

    NEWLINE

    MOV AH,2
    MOV DL,BL
    SUB DL,48
    INT 21H




    MAIN ENDP
END MAIN
```

ADD THREE NUMBER

```
.MODEL SMALL
.STACK 100H
.DATA

.CODE

NEWLINE MACRO  ; CREATE FUNCTION FOR NEWLINE
    MOV AH,2
    MOV DL,10
    INT 21H
    MOV DL,13
    INT 21H
ENDM
```

```
MAIN PROC
   MOV AH,1
   INT 21H
   MOV BL,AL

   NEWLINE

   MOV AH,1
   INT 21H
   MOV BH,AL

   NEWLINE

   MOV AH,1
   INT 21H
   MOV CL,AL

   ADD BL,BH
   SUB BL,48
   ADD BL,CL
   SUB BL,48

   NEWLINE

   MOV AH,2
   MOV DL,BL
   INT 21H



   MAIN ENDP
END MAIN
```

SUB TWO NUMBER

```
.MODEL SMALL
.STACK 100H
.DATA

 A DW 'FIRST NUMBER : $'
 B DW 'SECOND NUMER : $'
 C DW 'AFTER SUBSTRCTION : $'

.CODE

NEWLINE MACRO  ; CREATE FUNCTION FOR NEWLINE
    MOV AH,2
    MOV DL,10
    INT 21H
    MOV DL,13
    INT 21H
ENDM

MAIN PROC
    MOV AX,@DATA
    MOV DS,AX

    MOV AH,9   ; PRINT
    LEA DX,A
    INT 21H

    MOV AH,1     ;INPUT  1
    INT 21H
    MOV BL,AL


    NEWLINE

    MOV AH,9   ; PRINT
    LEA DX,B
    INT 21H
    MOV AH,1     ;INPUT 2
    INT 21H
    MOV BH,AL

    NEWLINE

    SUB BL,BH
    ADD BL,48

    MOV AH,9   ; PRINT
    LEA DX,C
    INT 21H

    MOV AH,2
    MOV DL,BL
    INT 21H

    MAIN ENDP
END MAIN
```

NOTE: MULTIPLY korar somoy al ar value ar sathe multiply hoi.bugfol al a thake. Input nile sub, code a number nile add 48 kore decimal nite hoi.

Multiply two number with Static initialization

```
.MODEL SMALL
.STACK 100H
.DATA

.CODE

NEWLINE MACRO  ; CREATE FUNCTION FOR NEWLINE
    MOV AH,2
    MOV DL,10
    INT 21H
    MOV DL,13
    INT 21H
ENDM

MAIN PROC
    MOV AL,3
    MOV BL,2
    MUL BL      ; AL=BL*AL

    MOV AH,2
    MOV DL,AL
    ADD DL,48
    INT 21H

    MAIN ENDP
END MAIN
```

Division of two number with Static initialization

```
.MODEL SMALL
.STACK 100H
.DATA

.CODE

NEWLINE MACRO  ; CREATE FUNCTION FOR NEWLINE
    MOV AH,2
    MOV DL,10
    INT 21H
    MOV DL,13
    INT 21H
ENDM

MAIN PROC
    MOV AL,6
    MOV BL,2
    DIV BL          ;AL/BL-- VAGFOL AL,VAGSES AH

    MOV CL,AL
    ADD CL,48
    MOV CH,AH
    ADD CH,48

    MOV AH,2
    MOV DL,CL
    INT 21H

    NEWLINE

    MOV AH,2
    MOV DL,CH
    INT 21H

    MAIN ENDP
END MAIN
```

IF CONDITION

```
.MODEL
.STACK 10H
.DATA


.CODE
```

```
NEWLINE MACRO
    MOV AH,2
    MOV DL,10
    INT 21H
    MOV DL,13
    INT 21H
ENDM


MAIN PROC
    MOV AH,1
    INT 21H
    MOV BL,AL

    NEWLINE

    MOV AH,1
    INT 21H
    MOV CL,AL

    CMP BL,CL       ; IF BL>CL PRINT BL
    JLE L1

    NEWLINE

    MOV AH,2
    MOV DL,BL
    INT 21H

    L1:

    MAIN ENDP
END MAIN
```

IF - ELSE CONDITION

```
.MODEL STACK
.STACK 100H

.DATA

.CODE

MAIN PROC
   MOV AX,10

   CMP AX,5
   JG L1      ; AX>5
   ADD AX,5
   JMP L2



   L1:
   SUB AX,5
   L2:
   MAIN ENDP
END MAIN
```

IF - IFELSE ELSE CONDITION

```
.MODEL STACK
.STACK 100H

.DATA
 A DW 'GRATER THAN $'
 B DW 'LESS THAN $'
 C DW ' EQUAL $'
```

```
.CODE

MAIN PROC
    MOV AX,@DATA
    MOV DS,AX

    MOV BX,10

    CMP BX,5
    JG L1
    JL L2
    MOV AH,9
    LEA DX,C
    INT 21H
    JMP L3



    L1:
    MOV AH,9
    LEA DX,A
    INT 21H
    JMP L3


    L2:
    MOV AH,9
    LEA DX,B
    INT 21H
    JMP L3

    L3:


    MAIN ENDP
```

```
END MAIN
```

AND (IF (A>=’A’ & A,=’Z’) PRINT(‘CAPITAL’)
ELSE PRINT (‘LOWER’))

```
.MODEL STACK
.STACK 100H

.DATA
 A DW 'CAPITAL $'
 B DW 'LOWER $'


.CODE

MAIN PROC
    MOV AX,@DATA
    MOV DS,AX

    MOV AL,'A'
    CMP AL,'A'
    JL L1
    CMP AL,'Z'
    JG L1

    MOV AH,9
    LEA DX,A
    INT 21H
    JMP L2



    L1:
    MOV AH,9
    LEA DX,B
    INT 21H

    L2:
```

```
   MAIN ENDP
END MAIN
```

OR

```
(  IF ( BL> 5 || BL =7)
PRINT(M)
ELSE
PRINT(N) )
```

```
.MODEL STACK
.STACK 100H

.DATA
 A DW 'M $'
 B DW 'N $'


.CODE

MAIN PROC
   MOV AX,@DATA
   MOV DS,AX

   MOV BL,21
   CMP BL,5
   JG L1:
   CMP BL,7
   JE L1

   MOV AH,9
   LEA DX,B
   INT 21H
   JMP L2
```

```
    L1:
    MOV AH,9
    LEA DX,A
    INT 21H

    L2:


    MAIN ENDP
END MAIN
```