

TUTORIAL 01:

```
.MODEL SMALL
.STACK 64
.DATA
    COUNT EQU 05
    DATA1 DB 125,235,197,91,48
    SUM DW ?
.CODE
    MOV AX,@DATA
    MOV DS,AX
    MOV CX,COUNT
    MOV SI,OFFSET DATA1
    MOV AX,0000H
BACK:MOV BL,[SI]
    CMP BL,50;comparision
    JB OVER
    CMP BL,150;comparision
    JA OVER
    ADD AL,[SI];addition
    JNC OVER
    INC AH
OVER: INC SI
    DEC CX
    JNZ BACK
    ;decimal display
    MOV BX,0000
    MOV CX,0AH
DCE:MOV DX,0000H
    DIV CX
    ADD DX,30H
    PUSH DX
    INC BX
    CMP AX,0000
    JA DCE
    MOV CX,BX
    MOV AH,02
SHOW:POP DX
    INT 21H
    LOOP SHOW
    MOV AH,4CH
    INT 21H
```

END

```
C:\MASM611\BIN>t1
216_
```

TUTORIAL 02:

```
.MODEL SMALL
.STACK 64
.DATA
    COUNT EQU 05
    DATA1 DB 125,235,197,91,48
    SUM DW ?
    XYZ DB ?
```

.CODE

```
    MOV AX,@DATA
    MOV DS,AX
    MOV CX,COUNT
    MOV SI,OFFSET DATA1
    MOV AX,0000H
BACK:MOV BL,[SI]
    CMP BL,30;comparision
    JB OVER
    CMP BL,100;comparision
    JA OVER
    ADD AL,[SI] ];addition
    JNC OVER
    INC AH
OVER: INC SI
    DEC CX
    JNZ BACK
    ;decimal display
    MOV BX,0000
    MOV CX,10H
DCE:MOV DX,0000H
    DIV CX
    CMP DX,000AH
    JA L1
    ADD DX,30H
    JMP L2
L1:ADD DX,37H
L2:PUSH DX
    INC BX
    CMP AX,0000
    JA DCE
    MOV CX,BX
    MOV AH,02
SHOW:POP DX
    INT 21H
    LOOP SHOW
    CALL EXIT
    EXIT PROC
    MOV AH,4CH
    INT 21H
    RET
    EXIT ENDP
```

END

```
C:\MASM611\BIN>t2
8B
C:\MASM611\BIN>
```

TUTORIAL 03:

```
.MODEL SMALL
.STACK 100
.DATA
    MSG DB ' ',10 DUP (?)
.CODE
```

```

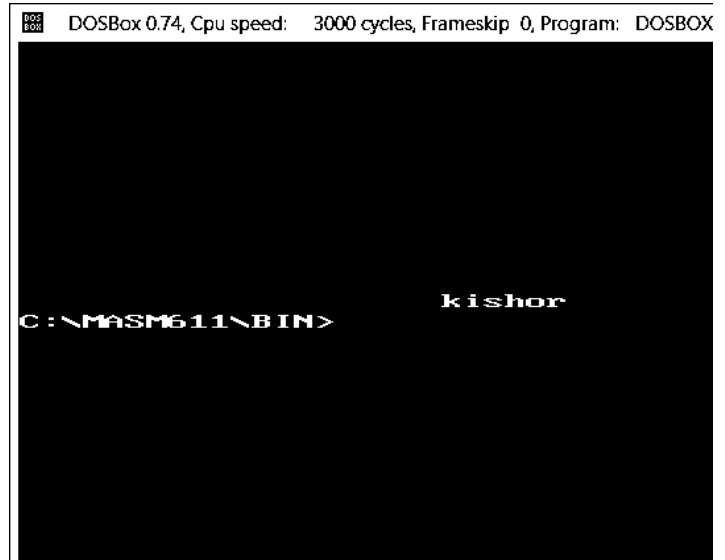
MOV AX,@DATA
MOV DS,AX
MOV AH,0AH
MOV CX,06H
LEA DX,MSG
INT 21H;string input
MOV SI,OFFSET MSG
MOV [SI+0008],'$'
CALL VIDEO_MODE
CALL CLEAR_SCREEN
CALL SET_CURSOR
CALL DISPLAY
CALL EXIT_PROGRAM
VIDEO_MODE PROC
MOV AH,00H
MOV AL,00H
INT 10H;set video mode
RET
VIDEO_MODE ENDP
CLEAR_SCREEN PROC
MOV AH,06H
MOV AL,00H
MOV BH,07H
MOV CX,0000H
MOV DX,1827H
INT 10H;clear whole screen
RET
CLEAR_SCREEN ENDP
SET_CURSOR PROC
MOV AH,02H
MOV DH,12
MOV DL,20
MOV BH,00
INT 10H;set cursor at centre
RET
SET_CURSOR ENDP
DISPLAY PROC
LEA DX,MSG
ADD DX,02H
MOV AH,09H
INT 21H;dispay string
RET
DISPLAY ENDP
EXIT_PROGRAM PROC
MOV AH,4CH
INT 21H
RET
EXIT_PROGRAM ENDP
END

```

```

C:\MASM611\BIN>t3
kishor

```



TUTORIAL 04:

```

.MODEL SMALL
.STACK 100
.DATA
    MSG DB ' ',10 DUP (?)
.CODE
    MOV AX,@DATA
    MOV DS,AX
    MOV AH,0AH
    MOV CX,06H
    LEA DX,MSG
    INT 21H;string input
    MOV SI,OFFSET MSG
    MOV [SI+0008],'$'
    MOV CX,0010
BACK:MOV AL,[SI]
    CMP AL,61H
    JB OVER
    CMP AL,7AH
    JA OVER
    AND AL,11011111B;convert to uppercase
OVER:MOV [SI],AL
    INC SI
    LOOP BACK
    CALL VIDEO_MODE
    CALL CLEAR_SCREEN
    CALL SET_CURSOR
    CALL DISPLAY
    CALL EXIT_PROGRAM
VIDEO_MODE PROC
MOV AH,00H
MOV AL,00H
INT 10H;set video mode
RET
VIDEO_MODE ENDP

CLEAR_SCREEN PROC

```

```

MOV AH,06H
MOV AL,00H
MOV BH,07H
MOV CX,0000H
MOV DX,1827H
INT 10H;clear whole screen
RET
CLEAR_SCREEN ENDP
SET_CURSOR PROC
MOV AH,02H
MOV DH,12
MOV DL,20
MOV BH,00
INT 10H;set cursor at centre
RET
SET_CURSOR ENDP
DISPLAY PROC
LEA DX,MSG
ADD DX,02H
MOV AH,09H
INT 21H;display string
RET
DISPLAY ENDP
EXIT_PROGRAM PROC
MOV AH,4CH
INT 21H
RET
EXIT_PROGRAM ENDP
END

```

C:\MASM611\BIN>t4
kishOr

DOSBox 0.74, CPU speed: 3000 cycles, Frameskip: 0, Program: DOSBOX

C:\MASM611\BIN>_ KISHOR

TUTORIAL 05:

```

.MODEL SMALL
.STACK 100
.DATA
    MSG DB ' ',10 DUP (?)
.CODE
    MOV AX,@DATA
    MOV DS,AX
    MOV AH,0AH

```

```

MOV CX,06H
LEA DX,MSG
INT 21H;input string
MOV SI,OFFSET MSG
MOV [SI+0008],'$'
CALL VIDEO_MODE
CALL CLEAR_SCREEN
CALL SET_CURSOR
CALL DISPLAY
CALL EXIT_PROGRAM
VIDEO_MODE PROC
MOV AH,00H
MOV AL,00H
INT 10H;set video mode
RET
VIDEO_MODE ENDP
CLEAR_SCREEN PROC
MOV AH,06H
MOV AL,00H
MOV BH,07H
MOV CX,0000H
MOV DX,1827H
INT 10H;clear whole screen
RET
CLEAR_SCREEN ENDP
SET_CURSOR PROC
MOV AH,02H
MOV DH,20
MOV DL,10
MOV BH,00
INT 10H;set cursor at centre
RET
SET_CURSOR ENDP
DISPLAY PROC
LEA DX,MSG
ADD DX,02
MOV AH,09H
INT 21H;dispay string
RET
DISPLAY ENDP
EXIT_PROGRAM PROC
MOV AH,4CH
INT 21H
RET
EXIT_PROGRAM ENDP
END

```

C:\MASM611\BIN>t5
joshi



```
C:\MASM611\BIN>
```

TUTORIAL 06:

```
.MODEL SMALL
.STACK 64
.DATA
    DATA DB 'mY NamE is KiShor', '$'
.CODE
    MOV AX,@DATA
    MOV DS,AX
    MOV SI,OFFSET DATA
    MOV CX,0017
BACK:MOV AL,[SI]
    CMP AL,61H
    JB OVER
    CMP AL,7AH
    JA OVER
    AND AL,11011111B;convert to upper case
OVER:MOV [SI],AL
    INC SI
    LOOP BACK
    CALL VIDEO_MODE
    CALL CLEAR_SCREEN
    CALL SET_CURSOR
    CALL DISPLAY
    CALL EXIT_PROGRAM
VIDEO_MODE PROC
    MOV AH,00H
    MOV AL,00H
    INT 10H;set video mode
    RET
VIDEO_MODE ENDP
CLEAR_SCREEN PROC
    MOV AH,06H
    MOV AL,00H
    MOV BH,07H
    MOV CX,0000H
    MOV DX,1827H
    INT 10H;clear whole screen
    RET
CLEAR_SCREEN ENDP
SET_CURSOR PROC
    MOV AH,02H
    MOV DH,12
```

```
MOV DL,20
MOV BH,00
INT 10H;set cursor at centre
RET
SET_CURSOR ENDP
DISPLAY PROC
    LEA DX,DATA
    MOV AH,09H
    INT 21H;dispay string
    RET
DISPLAY ENDP
EXIT_PROGRAM PROC
    MOV AH,4CH
    INT 21H
    RET
EXIT_PROGRAM ENDP
END
```



```
~BIN> MY NAME IS KISHOR
```

TUTORIAL 07:

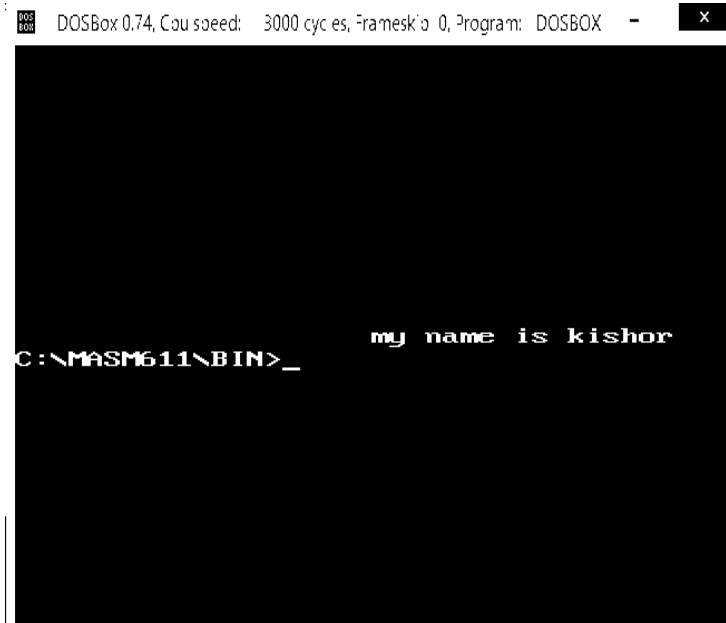
```
.MODEL SMALL
.STACK 64
.DATA
    DATA DB 'mY NamE is KiShor', '$'
.CODE
    MOV AX,@DATA
    MOV DS,AX
    MOV SI,OFFSET DATA
    MOV CX,0017
BACK:MOV AL,[SI]
    CMP AL,41H
    JB OVER
    CMP AL,5AH
    JA OVER
    OR AL,00100000B;convert to lowercase
OVER:MOV [SI],AL
    INC SI
    LOOP BACK
    CALL VIDEO_MODE
    CALL CLEAR_SCREEN
    CALL SET_CURSOR
```

```

CALL DISPLAY
CALL EXIT_PROGRAM
VIDEO_MODE PROC
MOV AH,00H
MOV AL,00H
INT 10H;set video mode
RET
VIDEO_MODE ENDP
CLEAR_SCREEN PROC
MOV AH,06H
MOV AL,00H
MOV BH,07H
MOV CX,0000H
MOV DX,1827H
INT 10H;clear whole screen
RET
CLEAR_SCREEN ENDP
SET_CURSOR PROC
MOV AH,02H
MOV DH,12
MOV DL,20
MOV BH,00
INT 10H;set cursor at centre
RET
SET_CURSOR ENDP
DISPLAY PROC
LEA DX,DATA
MOV AH,09H
INT 21H;display string
RET
DISPLAY ENDP
EXIT_PROGRAM PROC
MOV AH,4CH
INT 21H
RET
EXIT_PROGRAM ENDP
END

```

```
C:\MASM611\BIN>t7_
```



TUTORIAL 08:

```
.MODEL SMALL
```

```
.STACK 64
```

```
.DATA
```

```
SUM DW 0000
```

```
CARRY DW 0000
```

```
XYZ DB ?
```

```
.CODE
```

```
MOV AX,@DATA
```

```
MOV DS,AX
```

```
MOV CX,0064H
```

```
MOV AX,01
```

```
MOV BX,03
```

```
ADD AX,BX
```

```
MOV SUM,AX
```

```
BACK:MOV DX,AX
```

```
MOV AX,BX
```

```
MOV BX,DX
```

```
ADD SUM,BX
```

```
JNC NEXT
```

```
INC CARRY
```

```
NEXT:ADD AX,BX;addition of series
```

```
LOOP BACK
```

```
MOV AX,CARRY
```

```
CALL DISP
```

```
MOV AX,SUM
```

```
CALL DISP
```

```
CALL EXIT
```

```
DISP PROC;hex display
```

```
MOV BX,0000
```

```
MOV CX,10H
```

```
DCE:MOV DX,0000H
```

```
DIV CX
```

```
CMP DX,000AH
```

```
JA L1
```

```
ADD DX,30H
```

```

    JMP L2
L1:ADD DX,37H
L2:PUSH DX
    INC BX
    CMP AX,0000
    JA DCE
    MOV CX,BX
    MOV AH,02
SHOW:POP DX
    INT 21H
    LOOP SHOW
    RET
    DISP ENDP
    EXIT PROC
    MOV AH,4CH
    INT 21H
    RET
    EXIT ENDP

```

END

```

C:\MASM611\BIN>t8
2B8C6C

```

TUTORIAL 09:

.MODEL SMALL

.STACK 64

.DATA

SUM DW 0000

.CODE

```

    MOV AX,@DATA
    MOV DS,AX
    MOV AH,01H
    INT 21H
    SUB AL,30H
    MOV CH,00H
    MOV CL,AL
    MOV AH,02H
    MOV DL,20H
    INT 21H
    MOV AX,01
    MOV BX,03
    ADD AX,BX
    MOV SUM,AX

```

```

BACK:MOV DX,AX
    MOV AX,BX
    MOV BX,DX
    ADD SUM,BX
    ADD AX,BX;addition of series
    LOOP BACK
    MOV AX,SUM;decimal display
    MOV BX,0000
    MOV CX,0AH
DCE:MOV DX,0000H
    DIV CX

```

```

    ADD DX,30H
    PUSH DX
    INC BX
    CMP AX,0000
    JA DCE
    MOV CX,BX
    MOV AH,02

```

```

SHOW:POP DX
    INT 21H
    LOOP SHOW
    MOV AH,4C
    INT 21H

```

END

```

C:\MASM611\BIN>t9
9 518

```

TUTORIAL 10:

.MODEL SMALL

.STACK 100

.DATA

.CODE

```

    MOV AX,@DATA
    MOV DS,AX
    CALL VIDEO_MODE
    CALL CLEAR_SCREEN
    CALL SET_CURSOR
    CALL DISPLAY
    CALL EXIT_PROGRAM
    VIDEO_MODE PROC
    MOV AH,00H
    MOV AL,00H
    INT 10H;set video mode
    RET
    VIDEO_MODE ENDP
    CLEAR_SCREEN PROC
    MOV AH,06H
    MOV AL,00H
    MOV BH,0BFH
    MOV CX,0000H
    MOV DX,1827H
    INT 10H;clear whole screen
    RET
    CLEAR_SCREEN ENDP
    SET_CURSOR PROC
    MOV AH,02H
    MOV DH,10
    MOV DL,05
    MOV BH,00
    INT 10H;set cursor at centre
    RET
    SET_CURSOR ENDP
    DISPLAY PROC;display characters
    MOV CX,5FH

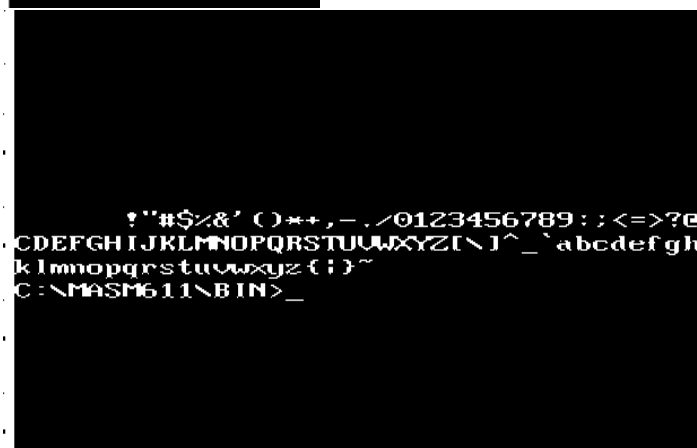
```

```

MOV BH,32
LABEL1:MOV AH,02H
MOV DL,BH
INT 21H
INC BH
LOOP LABEL1
RET
DISPLAY ENDP
EXIT_PROGRAM PROC
MOV AH,4CH
INT 21H
RET
EXIT_PROGRAM ENDP
END

```

C:\MASM611\BIN>t10



TUTORIAL 11:

```

.MODEL SMALL
.STACK 100
.DATA
    COUNT DB ?
    DATA1 DB 'kishor'
    DATA2 DW 5 DUP(?)
    SUM DW ?
    TEMP DW ?
    TEMP1 DW ?
    HUN DB 00
    TEN DB 00
.CODE
    MOV AX,@DATA
    MOV DS,AX
    MOV AH,06H;clear screen
    MOV AL,00H
    MOV BH,07H
    MOV CX,0000H
    MOV DX,1827H
    INT 10H;dos function
    LEA SI,DATA1
    LEA DI,DATA2;for storing BCD eqvt
    MOV DX,0000H
    MOV COUNT,05H

```

```

BACK:MOV HUN,00
    MOV TEN,00
    MOV AH,00H
    MOV AL,[SI]
L1: CMP AX,64H
    JB NEXT1
    INC HUN
    SUB AX,64H
    JMP L1
NEXT1: CMP AX,0AH
    JB NEXT2
    INC TEN
    SUB AX,0AH
    JMP NEXT1
NEXT2: MOV TEMP1,AX
    MOV AX,0001H
    MUL HUN
    MOV BX,100H
    MUL BX
    MOV TEMP,AX
    MOV AX,0001H
    MUL TEN
    MOV BX,10H
    MUL BX
    ADD AX,TEMP
    ADD AX,TEMP1
    MOV [DI],AX;storing BCD value of data
    INC SI
    INC DI
    INC DI
    DEC COUNT
    JNZ BACK
    LEA SI,DATA1
    MOV HUN,05
BACK2: MOV AL,[SI] ;display BCD value of data
    MOV AH,00
    MOV BX,0000
    MOV CX,0AH
DCE: MOV DX,0000H
    DIV CX
    ADD DX,30H
    PUSH DX
    INC BX
    CMP AX,0000
    JA DCE
    MOV CX,BX
    MOV AH,02
SHOW: POP DX
    INT 21H
    LOOP SHOW
    MOV AH,02
    MOV DL,20H

```

```

INT 21H
INC SI
DEC HUN
JNZ BACK2
MOV AH,4CH
INT 21H
END

```

```
C:\MASM611\BIN>t11_
```

```
107 105 115 104 111
```

TUTORIAL 12:

```

.MODEL SMALL
.STACK 64
.DATA
    MSG DB 10 DUP(?)
.CODE
    MOV AX,@DATA
    MOV DS,AX
    MOV AH,0AH
    MOV CX,06H
    LEA DX,MSG
    INT 21H;string input
    MOV SI,OFFSET MSG
    MOV CX,0006H
    MOV BX,0000H
    ADD SI,02H
BACK:MOV AH,[SI]
    CMP AH,61H;compairing vowel
    JE NEXT
    CMP AH,65H;compairing vowel
    JE NEXT
    CMP AH,69H;compairing vowel
    JE NEXT
    CMP AH,6FH;compairing vowel
    JE NEXT
    CMP AH,75H;compairing vowel
    JE NEXT
    JMP GOTO1
NEXT:INC BX;count vowel
GOTO1:INC SI
    LOOP BACK
    MOV AH,02
    MOV DL,0AH
    INT 21H;print space
;Display count in BCD
    MOV AX,BX
    MOV BX,0000
    MOV CX,0AH
DCE:MOV DX,0000H
    DIV CX
    ADD DX,30H
    PUSH DX

```

```

INC BX
CMP AX,0000
JA DCE
MOV CX,BX
MOV AH,02
SHOW:POP DX
INT 21H
LOOP SHOW
MOV AH,4CH
INT 21H

```

END

```
C:\MASM611\BIN>t12
kishor
2
```

TUTORIAL 13:

```

.MODEL SMALL
.STACK 64
.DATA
    MSG DB 10 DUP(?)
.CODE
    MOV AX,@DATA
    MOV DS,AX
    MOV AH,0AH
    MOV CX,06H
    LEA DX,MSG
    INT 21H;input string
    MOV SI,OFFSET MSG
    MOV CX,0006H
    MOV BX,0000H
    ADD SI,02H
BACK:MOV AH,[SI]
    CMP AH,61H;compairing vowel
    JE NEXT
    CMP AH,65H;compairing vowel
    JE NEXT
    CMP AH,69H;compairing vowel
    JE NEXT
    CMP AH,6FH;compairing vowel
    JE NEXT
    CMP AH,75H;compairing vowel
    JE NEXT
    JMP GOTO1
NEXT:AND AH,11011111B;convert to upppercase
    MOV [SI],AH
GOTO1:INC SI
    LOOP BACK
    MOV [SI],'$'
    CALL VIDEO_MODE
    CALL CLEAR_SCREEN
    CALL SET_CURSOR
    CALL DISPLAY
    CALL EXIT_PROGRAM

```



```

VIDEO_MODE PROC
MOV AH,00H
MOV AL,00H
INT 10H;set video mode
RET
VIDEO_MODE ENDP
CLEAR_SCREEN PROC
MOV AH,06H
MOV AL,00H
MOV BH,07H
MOV CX,0000H
MOV DX,1827H
INT 10H;clear screen
RET
CLEAR_SCREEN ENDP
SET_CURSOR PROC
MOV AH,02H
MOV DH,12
MOV DL,20
MOV BH,00
INT 10H;set cursor at centre
RET
SET_CURSOR ENDP
DISPLAY PROC
LEA DX,MSG
ADD DX,02H
MOV AH,09H
INT 21H;display string
RET
DISPLAY ENDP
EXIT_PROGRAM PROC
MOV AH,4CH
INT 21H
RET
EXIT_PROGRAM ENDP
END

```

```

C:\MASM611\BIN>t13
aedgio_

```

AEdg IO

TUTORIAL 14:

.MODEL SMALL

.STACK 64

.DATA

MSG DB 10 DUP(?)

.CODE

```

MOV AX,@DATA
MOV DS,AX
MOV AH,0AH
MOV CX,06H
LEA DX,MSG

```

```

INT 21H;string input
MOV SI,OFFSET MSG
ADD SI,02H
MOV CX,0006H
BACK:MOV AL,[SI]
CMP AL,61H
JB OVER
CMP AL,7AH
JA OVER
AND AL,11011111B;convert to uppercase
OVER:MOV [SI],AL
INC SI
LOOP BACK
MOV [SI],'$'
CALL VIDEO_MODE
CALL CLEAR_SCREEN
CALL SET_CURSOR
CALL DISPLAY
CALL EXIT_PROGRAM
VIDEO_MODE PROC
MOV AH,00H
MOV AL,00H
INT 10H;set video mode
RET
VIDEO_MODE ENDP
CLEAR_SCREEN PROC
MOV AH,06H
MOV AL,00H
MOV BH,07H
MOV CX,0000H
MOV DX,1827H
INT 10H;clear screen
RET
CLEAR_SCREEN ENDP
SET_CURSOR PROC
MOV AH,02H
MOV DH,12
MOV DL,20
MOV BH,00
INT 10H;set cursor at co ordinate
RET
SET_CURSOR ENDP
DISPLAY PROC
LEA DX,MSG
ADD DX,02H
MOV AH,09H
INT 21H;display string
RET
DISPLAY ENDP
EXIT_PROGRAM PROC
MOV AH,4CH
INT 21H

```

```
RET
EXIT_PROGRAM ENDP
END
```

```
C:\MASM611\BIN>t14
kishor
```

```
C:\MASM611\BIN>

KISHOR

C:\MASM611\BIN>
```

TUTORIAL 15:

```
.MODEL SMALL
.STACK 64
.DATA
    MSG DB 10 DUP(?)
.CODE
    MOV AX,@DATA
    MOV DS,AX
    MOV AH,0AH
    MOV CX,06H
    LEA DX,MSG
    INT 21H;input string
    MOV SI,OFFSET MSG
    ADD SI,02H
    MOV CX,0006H
BACK:MOV AL,[SI]
    CMP AL,41H
    JB OVER
    CMP AL,5AH
    JA OVER
    OR AL,00100000B;convert to lowercase
OVER:MOV [SI],AL
    INC SI
    LOOP BACK
    MOV [SI],'$'
    CALL VIDEO_MODE
    CALL CLEAR_SCREEN
    CALL SET_CURSOR
    CALL DISPLAY
    CALL EXIT_PROGRAM
VIDEO_MODE PROC
    MOV AH,00H
    MOV AL,00H
    INT 10H;set video mode
    RET
VIDEO_MODE ENDP
CLEAR_SCREEN PROC
    MOV AH,06H
    MOV AL,00H
    MOV BH,07H
    MOV CX,0000H
    MOV DX,1827H
    INT 10H;clear screen
```

```
RET
CLEAR_SCREEN ENDP
SET_CURSOR PROC
    MOV AH,02H
    MOV DH,10
    MOV DL,02
    MOV BH,00
    INT 10H;set cursor at co ordinate
    RET
SET_CURSOR ENDP
DISPLAY PROC
    LEA DX,MSG
    ADD DX,02H
    MOV AH,09H
    INT 21H;display string
    RET
DISPLAY ENDP
EXIT_PROGRAM PROC
    MOV AH,4CH
    INT 21H
    RET
EXIT_PROGRAM ENDP
END
```

```
C:\MASM611\BIN>t15
KISHOR
```

```

kishor
C:\MASM611\BIN>_
```

TUTORIAL 16:

```
.MODEL SMALL
.STACK 200
.DATA
    MSG DB 'KISHOR$'
    TEMP1 DW 65535;for delay
    TEMP2 DW 65535;for delay
    TEMP3 DW 65535;for delay
    TEMP4 DW 65535;for delay
    TEMP5 DW 65535;for delay
    LOC DW 0021H
.CODE
    MOV AX,@DATA
    MOV DS,AX
    CALL VIDEO_MODE
    MOV CX,25
    BACK:CALL CLEAR_SCREEN
```



```

MOV DX,LOC
CALL SET_CURSOR
CALL DISPLAY
BACK3:DEC TEMP3;delay loop
JNZ BACK3
BACK1:DEC TEMP1;delay loop
JNZ BACK1
BACK2:DEC TEMP2;delay loop
JNZ BACK2
BACK4:DEC TEMP4;delay loop
JNZ BACK4
BACK5:DEC TEMP5;delay loop
JNZ BACK5
SUB LOC,02
LOOP BACK
CALL EXIT_PROGRAM
VIDEO_MODE PROC
MOV AH,00H
MOV AL,00H
INT 10H;set video mode
RET
VIDEO_MODE ENDP
CLEAR_SCREEN PROC
MOV AH,06H
MOV AL,00H
MOV BH,07H
MOV CX,0000H
MOV DX,1827H
INT 10H;clear screen
RET
CLEAR_SCREEN ENDP
SET_CURSOR PROC
MOV AH,02H
MOV BH,00
INT 10H;set cursor
RET
SET_CURSOR ENDP
DISPLAY PROC
MOV AH,09H
LEA DX,MSG
INT 21H;display string
RET
DISPLAY ENDP
EXIT_PROGRAM PROC
MOV AH,4CH
INT 21H
RET
EXIT_PROGRAM ENDP
END

```

**TUTORIAL****17:**

.MODEL SMALL

.STACK 64

.DATA

NUM DB 2 DUP(?)

TEMP DW ?

TEMP2 DW ?

TEMP3 DW ?

TEMP4 DW ?

.CODE

MOV AX,@DATA

MOV DS,AX

MOV CX,0002

LEA DI,NUM

KJ:MOV AH,01H;input two digit

INT 21H

MOV [DI],AL

INC DI

LOOP KJ

MOV AH,02H

MOV DL,20H

INT 21H

MOV SI,OFFSET NUM

MOV DH,[SI]

INC SI

MOV DL,[SI]

SUB DH,30H;converting to eqvt HEX

SUB DL,30H

MOV CL,04H

ROL DH,CL

OR DH,DL

MOV CL,DH

MOV CH,00H

LABELS:CMP CL,10H

JB NEXT

INC CH

SUB CL,10H

JMP LABELS

NEXT:ADD CL,0AH

DEC CH

JNZ NEXT

MOV CH,00

MOV TEMP,CX;HEX eqvt to temp

CALL VIDEO_MODE

CALL CLEAR_SCREEN

MOV CX,000AH

MOV BL,01H

MOV DH,00H

```

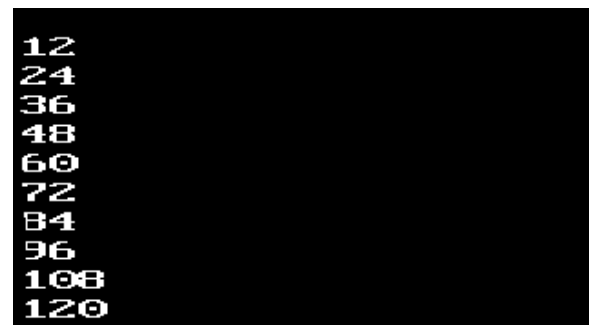
MOV DL,00H;set cursor at top at first
LABEL1:CALL SET_CURSOR
MOV AX,TEMP
MUL BL;multiplied at ax
CALL DECIMAL
INC BL
LOOP LABEL1
MOV AH,4CH
INT 21H
VIDEO_MODE PROC
MOV AH,00H
MOV AL,00H
INT 10H;set video mode
RET
VIDEO_MODE ENDP
CLEAR_SCREEN PROC
MOV AH,06H
MOV AL,00H
MOV BH,07H
MOV CX,0000H
MOV DX,1827H
INT 10H;clear whole screen
RET
CLEAR_SCREEN ENDP
SET_CURSOR PROC
MOV AH,02H
MOV BH,00
INC DH
MOV DL,00
INT 10H;set cursor
RET
SET_CURSOR ENDP
DECIMAL PROC;display eqvt BCD
MOV TEMP4,BX
MOV BX,0000H
MOV TEMP3,CX
MOV CX,000AH
MOV TEMP2,DX
DCE: MOV DX,0000H
DIV CX
ADD DX,30H
PUSH DX
INC BX
CMP AX,0000
JA DCE
MOV CX,BX
MOV AH,02H
SHOW: POP DX
INT 21H
LOOP SHOW
MOV DX,TEMP2
MOV CX,TEMP3

```

```

MOV BX,TEMP4
RET
DECIMAL ENDP
END

```



TUTORIAL 18:

```

.MODEL SMALL
.STACK 64
.DATA
    NUM1 DW 0005
    NUM2 DW 0015
    TEMP1 DW ?
    TEMP2 DW ?
    SUM DW ?
.CODE
    MOV AX,@DATA
    MOV DS,AX
    MOV AX,NUM1
    MOV BX,NUM2
    CMP AX,BX
    JA NEXT
    XCHG AX,BX;find greatest
    MOV SUM,BX
NEXT:MOV DX,0000H;finding HCF
    MOV TEMP1,AX
    MOV TEMP2,BX
    DIV BX
    CMP DX,0000H
    JE LABEL1
    DEC TEMP2
    MOV AX,TEMP1
    MOV BX,TEMP2
    JMP NEXT
LABEL1:MOV AX,SUM
    MOV BX,TEMP2
    MOV DX,0000H
    DIV BX
    CMP DX,0000H
    JE LABEL2
    DEC TEMP2
    MOV AX,TEMP1
    MOV BX,TEMP2
    JMP NEXT
LABEL2:MOV AX,TEMP2;HCF

```

```

;BCD display
MOV BX,0000
MOV CX,0AH
DCE:MOV DX,0000H
DIV CX
ADD DX,30H
PUSH DX
INC BX
CMP AX,0000
JA DCE
MOV CX,BX
MOV AH,02
SHOW:POP DX
INT 21H
LOOP SHOW
MOV AH,4CH
INT 21H
END
C:\MASM611\BIN>t18
5
TUTORIAL 19:
.MODEL SMALL
.STACK 64
.DATA
    NUM1 DW 0005
    NUM2 DW 0003
    TEMP1 DW ?
    TEMP2 DW ?
    SUM DW ?
.CODE
    MOV AX,@DATA
    MOV DS,AX
    MOV AX,NUM1
    MOV BX,NUM2
    CMP AX,BX
    JA NEXT
    XCHG AX,BX;find greatest
    MOV SUM,BX
NEXT:MOV DX,0000H
    MOV TEMP1,AX
    MOV TEMP2,BX
    DIV BX
    CMP DX,0000H
    JE LABEL1
    DEC TEMP2
    MOV AX,TEMP1
    MOV BX,TEMP2
    JMP NEXT
LABEL1:MOV AX,SUM
    MOV BX,TEMP2
    MOV DX,0000H
    DIV BX

```

```

CMP DX,0000H
JE LABEL2
DEC TEMP2
MOV AX,TEMP1
MOV BX,TEMP2;HCF
JMP NEXT
LABEL2:MOV AX,NUM1
    MUL NUM2
    MOV DX,0000H
    DIV TEMP2;LCM
    BCD display
    MOV BX,0000
    MOV CX,0AH
DCE:MOV DX,0000H
    DIV CX
    ADD DX,30H
    PUSH DX
    INC BX
    CMP AX,0000
    JA DCE
    MOV CX,BX
    MOV AH,02
SHOW:POP DX
    INT 21H
    LOOP SHOW
    MOV AH,4CH
    INT 21H
END
C:\MASM611\BIN>t19
15
TUTORIAL 20:
.MODEL SMALL
.STACK 100
.DATA
    MSG DB 42 DUP(?)
    TEMP DB ?
.CODE
    MOV AX,@DATA
    MOV DS,AX
    LEA SI,MSG
    MOV [SI],41
    MOV AH,0AH
    MOV CX,0040
    LEA DX,MSG
    INT 21H;input string
    MOV SI,OFFSET MSG
    ADD SI,02
    MOV [SI+0041],'$'
    CALL VIDEO_MODE
    CALL CLEAR_SCREEN
    MOV DX,0205H
    MOV BL,00H

```

LABEL1:CALL SET_CURSOR;set cursor at every space

MOV AH,02H

MOV TEMP,DL

BACK:MOV DL,[SI]

CMP DL,20H;check space

JNE NEXT

INC DH

MOV DL,TEMP

INC DL

INC SI

JMP LABEL1

NEXT:CMP DL,'\$';check end point

JE GO1

INC SI

INC TEMP

INT 21H

JMP BACK

GO1:CALL EXIT_PROGRAM

VIDEO_MODE PROC

MOV AH,00H

MOV AL,02H

INT 10H;set video mode

RET

VIDEO_MODE ENDP

CLEAR_SCREEN PROC

MOV AH,06H

MOV AL,00H

MOV BH,07H

MOV CX,0000H

MOV DX,1827H

INT 10H;clear screen

RET

CLEAR_SCREEN ENDP

SET_CURSOR PROC

MOV AH,02H

MOV BH,00

INT 10H;set cursor as co ordinate

RET

SET_CURSOR ENDP

EXIT_PROGRAM PROC

MOV AH,4CH

INT 21H

RET

EXIT_PROGRAM ENDP

END

C:\MASM611\BIN>t20

programming in assembly language is fun_



DOSBox 0.74, Cpu speed: 3000 cycles, Frameskip

programming
in
assembly
language
is
fun

TUTORIAL 21:

.MODEL SMALL

.STACK 64

.DATA

NUM1 DB 1,2,3,4,5,6,7,8,9

NUM2 DB 1,2,3,4,5,6,7,8,9

NUM3 DB 9 DUP(?)

TEMP1 DB 3

TEMP2 DB 3

.CODE

MOV AX,@DATA

MOV DS,AX

LEA SI,NUM1

LEA BX,NUM2

LEA DI,NUM3

MOV CX,09

BACK:MOV AL,[SI]

ADD AL,[BX]

MOV [DI],AL

INC SI

INC DI

INC BX

LOOP BACK

LEA SI,NUM3

BACK2:MOV BX,0000

MOV CX,000AH

MOV AL,[SI]

MOV AH,00

DDD: MOV DX,0000

DIV CX

ADD DX,30H

PUSH DX

INC BX

CMP AX,0000

JA DDD

MOV CX,BX

MOV AH,02

SHOW:POP DX

INT 21H

LOOP SHOW

```

INC SI
MOV AH,02
MOV DL,20H
INT 21H
DEC TEMP1
JNZ BACK2
MOV TEMP1,03
MOV AH,02
MOV DL,0AH
INT 21H
DEC TEMP2
JNZ BACK2
MOV AH,4CH
INT 21H

```

END

```

C:\MASM611\BIN>t21
2 4 6
8 10 12
14 16 18

```

TUTORIAL 22:

```

.MODEL SMALL
.STACK 64
.DATA
    SUM DW ?
    NUM DB 2 DUP(?)
.CODE
    MOV AX,@DATA
    MOV DS,AX
    MOV CX,0002
    LEA DI,NUM
    KJ:MOV AH,01H;input two digit no
        INT 21H
        MOV [DI],AL
        INC DI
        LOOP KJ
        MOV AH,02H
        MOV DL,20H
        INT 21H;print space
        MOV SI,OFFSET NUM
        MOV DH,[SI]
        INC SI
        MOV DL,[SI]
        SUB DH,30H
        SUB DL,30H
        MOV CL,04H
        ROL DH,CL
        OR DH,DL
        MOV AX,0000H
        MOV CL,DH
        MOV CH,00H
    LABELS: CMP CL,10H
        JB NEXT

```

```

INC CH
SUB CL,10H
JMP LABELS
NEXT:ADD CL,0AH;converted to eqvt HEX
    DEC CH
    JNZ NEXT
BACK:ADD AX,CX
    LOOP BACK
    MOV SUM,AX;adding
    MOV AX,SUM;BCD display
    MOV BX,0000
    MOV CX,0AH
DCE:MOV DX,0000H
    DIV CX
    ADD DX,30H
    PUSH DX
    INC BX
    CMP AX,0000
    JA DCE
    MOV CX,BX
    MOV AH,02
SHOW:POP DX
    INT 21H
    LOOP SHOW
    MOV AH,02
    MOV DL,20H
    INT 21H;print space
    MOV AX,SUM;HEX display
    MOV BX,0000
    MOV CX,10H
DCE1:MOV DX,0000H
    DIV CX
    CMP DX,000AH
    JA L3
    ADD DX,30H
    JMP L2
L3:ADD DX,37H
L2:PUSH DX
    INC BX
    CMP AX,0000
    JA DCE1
    MOV CX,BX
    MOV AH,02
SHOW1:POP DX
    INT 21H
    LOOP SHOW1
    MOV AH,4C
    INT 21H

```

END

```

C:\MASM611\BIN>t22
15 120 78

```

TUTORIAL 23:

```

.MODEL SMALL
.STACK 64
.DATA
    SUM DW ?
    NUM DB 2 DUP(?)
.CODE
    MOV AX,@DATA
    MOV DS,AX
    MOV CX,0002
    LEA DI,NUM
KJ:MOV AH,01H;input 2 digit no
    INT 21H
    MOV [DI],AL
    INC DI
    LOOP KJ
    MOV AH,02H
    MOV DL,20H
    INT 21H
    MOV SI,OFFSET NUM
    MOV DH,[SI]
    INC SI
    MOV DL,[SI]
    SUB DH,30H
    SUB DL,30H
    MOV CL,04H
    ROL DH,CL
    OR DH,DL
    MOV AX,0000H
    MOV CL,DH
    MOV CH,00H
LABELS:CMP CL,10H
    JB NEXT
    INC CH
    SUB CL,10H
    JMP LABELS
NEXT:CMP CH,00H
    JA KJ
    JMP JK
KJ:ADD CL,0AH;converted to eqvt HEX
    DEC CH
    JNZ KJ;INPUT
JK:MOV BX,0000H
BACK:MOV AX,CX
    INC AX
    MOV DL,AL
    MUL DL
    ADD BX,AX;addition of series
    LOOP BACK
    MOV DX,0000H
    MOV AX,BX
    MOV CX,0002
    MUL CX

```

```

MOV SUM,AX
MOV CH,00H
MOV AX,SUM;decimal;display;of;sum
MOV BX,0000
MOV CX,0AH
DCE:MOV DX,0000H
    DIV CX
    ADD DX,30H
    PUSH DX
    INC BX
    CMP AX,0000
    JA DCE
    MOV CX,BX
    MOV AH,02
SHOW:POP DX
    INT 21H
    LOOP SHOW
    MOV AH,02
    MOV DL,20H
    INT 21H
    MOV AX,SUM;HEX;display
    MOV BX,0000
    MOV CX,10H
DCE1:MOV DX,0000H
    DIV CX
    CMP DX,000AH
    JA L3
    ADD DX,30H
    JMP L2
L3:ADD DX,37H
L2:PUSH DX
    INC BX
    CMP AX,0000
    JA DCE1
    MOV CX,BX
    MOV AH,02
SHOW1:POP DX
    INT 21H
    LOOP SHOW1
    MOV AH,4C
    INT 21H
END

```

C:\MASM611\BIN>t23
06 278 4E

```

TUTORIAL 24:
.MODEL SMALL
.STACK 100
.DATA
    TRUE DB 'THE NUMBER IS PRIME$'
    FALSE DB 'THE NUMBER IS NOT PRIME$'
    SUM DW ?
    NUM DB 2 DUP(?)

```



```

TEMP DW ?
.CODE
MOV AX,@DATA
MOV DS,AX
MOV CX,0002
LEA DI,NUM
KJ:MOV AH,01H
INT 21H
MOV [DI],AL
INC DI
LOOP KJ
MOV AH,02H
MOV DL,20H
INT 21H
MOV SI,OFFSET NUM
MOV DH,[SI]
INC SI
MOV DL,[SI]
SUB DH,30H
SUB DL,30H
MOV CL,04H
ROL DH,CL
OR DH,DL
MOV AX,0000H
MOV CL,DH
MOV CH,00H
LABELS:CMP CL,10H
JB NEXT
INC CH
SUB CL,10H
JMP LABELS
NEXT:CMP CH,00H
JA KJ
JMP JK
KJ:ADD CL,0AH
DEC CH
JNZ KJ;INPUT
JK:MOV AX,CX
MOV TEMP,AX
DEC CX
DEC CX
BACKA:MOV DX,CX
INC DX
DIV DL
CMP AH,00H
JE YES
MOV AX,TEMP
LOOP BACKA
JMP NO
YES:LEA DX,FALSE
MOV AH,09H
INT 21H

```

```

JMP EN
NO:LEA DX,TRUE
MOV AH,09H
INT 21H
EN:MOV AH,4CH
INT 21H
END

```

```

C:\MASM611\BIN>t24
07 THE NUMBER IS PRIME
C:\MASM611\BIN>t24
88 THE NUMBER IS NOT PRIME

```

TUTORIAL 25:

```

.MODEL SMALL
.STACK 100H
.DATA
MSG DB "TODAY'S DATE IS ',' '$'
MSG1 DB "TODAY'S TIME IS ',' '$'
.CODE
MOV AX,@DATA
MOV DS,AX
MOV DX,OFFSET MSG
MOV AH,09H
INT 21H
MOV AH,2AH
INT 21H
PUSH CX
MOV CX,0
MOV CL,DL
PUSH CX
MOV CL,DH
PUSH CX
MOV DH,0
MOV DX,0
POP AX
MOV CX,0
MOV BX,10
DIVIDE1: DIV BX
PUSH DX
ADD CX,1
MOV DX,0
CMP AX,0
JNE DIVIDE1
DIVIDE2: POP DX
ADD DL,30H
MOV AH,02H
INT 21H
LOOP DIVIDE2
MOV DL','
MOV AH,02H
INT 21H
MOV DX,0
POP AX

```

```

MOV CX,0
MOV BX,10
DIVIDE3: DIV BX
PUSH DX
ADD CX,1
MOV DX,0
CMP AX,0
JNE DIVIDE3
DIVIDE4: POP DX
ADD DL,30H
MOV AH,02H
INT 21H
LOOP DIVIDE4
MOV DL,'/'
MOV AH,02H
INT 21H
MOV DX,0
POP AX
MOV CX,0
MOV BX,10
DIVIDE5: DIV BX
PUSH DX
ADD CX,1
MOV DX,0
CMP AX,0
JNE DIVIDE5
DIVIDE6: POP DX
ADD DL,30H
MOV AH,02H
INT 21H
LOOP DIVIDE6
MOV DL,0AH
MOV AH,02H
INT 21H
MOV DX,OFFSET MSG1
MOV AH,09H
INT 21H
MOV AH,2CH
INT 21H
MOV DL,DH
MOV DH,0
PUSH DX
MOV DL,CL
PUSH DX
MOV DL,CH
PUSH DX
MOV DH,0
MOV DX,0
POP AX
MOV CX,0
MOV BX,10
TDIVIDE1: DIV BX

```

```

PUSH DX
ADD CX,1
MOV DX,0
CMP AX,0
JNE TDIVIDE1
TDIVIDE2: POP DX
ADD DL,30H
MOV AH,02H
INT 21H
LOOP TDIVIDE2
MOV DL,'-'
MOV AH,02H
INT 21H
MOV DX,0
POP AX
MOV CX,0
MOV BX,10
TDIVIDE3: DIV BX
PUSH DX
ADD CX,1
MOV DX,0
CMP AX,0
JNE TDIVIDE3
TDIVIDE4: POP DX
ADD DL,30H
MOV AH,02H
INT 21H
LOOP TDIVIDE4
MOV DL,'-'
MOV AH,02H
INT 21H
MOV DX,0
POP AX
MOV CX,0
MOV BX,10
TDIVIDE5: DIV BX
PUSH DX
ADD CX,1
MOV DX,0
CMP AX,0
JNE TDIVIDE5
TDIVIDE6: POP DX
ADD DL,30H
MOV AH,02H
INT 21H
LOOP TDIVIDE6
MOV AH,4CH
INT 21H
END

```

```

C:\MASM611\BIN>t25
TODAY'S DATE IS 7/30/2016
TODAY'S TIME IS 14-17-12

```