TUTORIAL 01:	.CODE
.MODEL SMALL	MOV AX,@DATA
.STACK 64	MOV DS,AX
.DATA	MOV CX,COUNT
COUNT EQU 05	MOV SI,OFFSET DATA1
DATA1 DB 125,235,197,91,48	MOV AX,0000H
SUM DW ?	BACK:MOV BL,[SI]
.CODE	CMP BL,30;comparision
MOV AX,@DATA	JB OVER
MOV DS,AX	CMP BL,100;comparision
MOV CX,COUNT	JA OVER
MOV SI,OFFSET DATA1	ADD AL,[SI]];addition
MOV AX,0000H	JNC OVER
BACK:MOV BL,[SI]	INC AH
CMP BL,50;comparision	OVER: INC SI
JB OVER	DEC CX
CMP BL,150; comparision	JNZ BACK
JA OVER	;decimal display
ADD AL,[SI];addition	MOV BX,0000
JNC OVER	MOV CX,10H
INC AH	DCE:MOV DX,0000H
OVER: INC SI	DIV CX
DEC CX	CMP DX,000AH
JNZ BACK	JA L1
;decimal display	ADD DX,30H
MOV BX,0000	JMP L2
MOV CX,0AH	L1:ADD DX,37H
DCE:MOV DX,0000H	L2:PUSH DX
DIV CX	INC BX
ADD DX,30H	CMP AX,0000
PUSH DX	JA DCE
INC BX	MOV CX,BX
CMP AX,0000	MOV AH,02
JA DCE	SHOW:POP DX
MOV CX,BX	INT 21H
MOV AH,02	LOOP SHOW
SHOW:POP DX	CALL EXIT
INT 21H	EXIT PROC
LOOP SHOW	MOV AH,4CH
MOV AH,4CH	INT 21H
INT 21H	RET
END	EXIT ENDP
C:\MASM611\BIN>t1	END
216_	
	C:\MASM611\BIN>t2
TUTORIAL 02:	8B C:\MASM611\BIN>
.MODEL SMALL	
.STACK 64	TUTORIAL 03:
.DATA	.MODEL SMALL
COUNT EQU 05	.STACK 100
DATA1 DB 125,235,197,91,48	.DATA
SUM DW ?	MSG DB ' ',10 DUP (?)
XYZ DB ?	.CODE

MOV AX,@DATA	DOSBox 0.74, Cpu speed: 3000 cycles, Frameskip 0, Programs
MOV DS,AX	
MOV AH,0AH	
MOV CX,06H	
LEA DX,MSG	
INT 21H;string input	
MOV SI,OFFSET MSG	
MOV [SI+0008],'\$'	kishor
CALL VIDEO_MODE	C:\MASM611\BIN>
CALL CLEAR SCREEN	
CALL SET_CURSOR	
CALL DISPLAY	
CALL EXIT_PROGRAM	
VIDEO_MODE PROC	
MOV AH,00H	
MOV AL,00H	TUTORIAL 04:
INT 10H;set video mode	.MODEL SMALL
RET	.STACK 100
VIDEO_MODE ENDP	.DATA
CLEAR SCREEN PROC	MSG DB ' ',10 DUP (?)
MOV AH,06H	.CODE
MOV AL,00H	MOV AX,@DATA
MOV BH,07H	MOV DS,AX
MOV CX,0000H	MOV AH,0AH
MOV DX,1827H	MOV CX,06H
INT 10H;clear whole screen	LEA DX,MSG
RET	INT 21H;string input
CLEAR_SCREEN ENDP	MOV SI,OFFSET MSG
SET_CURSOR PROC	MOV [SI+0008],'\$'
MOV AH,02H	MOV CX,0010
MOV DH,12	BACK:MOV AL,[SI]
MOV DL,20	CMP AL,61H
MOV BH,00	JB OVER
INT 10H;set cursor at centre	CMP AL,7AH
RET	JA OVER
SET_CURSOR ENDP	AND AL,11011111B;convert to uppercase
DISPLAY PROC	OVER:MOV [SI],AL
LEA DX,MSG	INC SI
ADD DX,02H	LOOP BACK
MOV AH,09H	CALL VIDEO_MODE
INT 21H;dispay string	CALL CLEAR_SCREEN
RET	CALL SET_CURSOR
DISPLAY ENDP	CALL DISPLAY
	CALL EXIT_PROGRAM
EXIT_PROGRAM PROC	VIDEO_MODE PROC
MOV AH,4CH	MOV AH,00H
INT 21H RET	MOV AL,00H
	INT 10H;set video mode
EXIT_PROGRAM ENDP	RET
END	VIDEO_MODE ENDP
C:\MASM611\BIN>t3	
kishor	CLEAR SCREEN PROC

CLEAR_SCREEN PROC

DOSBOX

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



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

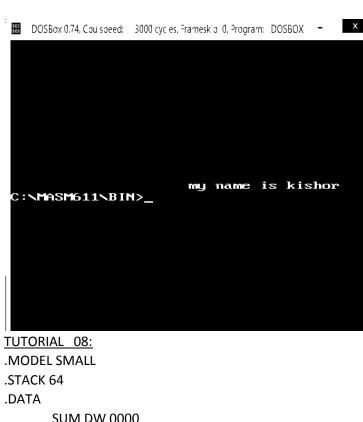
C:\MASM611\BIN>t5 joshi

	MOV DL,20
	MOV BH,00
	·
	INT 10H;set cursor at centre
	RET
	SET_CURSOR ENDP
	DISPLAY PROC
	LEA DX,DATA
	MOV AH,09H
	·
joshi	INT 21H;dispay string
C:\MASM611\BIN>	RET
TUTORIAL 06:	DISPLAY ENDP
.MODEL SMALL	EXIT_PROGRAM PROC
.STACK 64	MOV AH,4CH
.DATA	INT 21H
	RET
DATA DB 'mY NamE is KiShor','\$'	EXIT_PROGRAM ENDP
.CODE	END
MOV AX,@DATA	END
MOV DS,AX	
MOV SI,OFFSET DATA	
MOV CX,0017	
BACK:MOV AL,[SI]	
CMP AL,61H	
JB OVER	
	MY NAME IS KISHOR
CMP AL,7AH	∖BIN>
JA OVER	
AND AL,11011111B; convert to upper case	
OVER:MOV [SI],AL	
INC SI	
LOOP BACK	
CALL VIDEO_MODE	
CALL CLEAR_SCREEN	TUTORIAL 07:
CALL SET_CURSOR	.MODEL SMALL
CALL DISPLAY	.STACK 64
	.DATA
CALL EXIT_PROGRAM	
VIDEO_MODE PROC	DATA DB 'mY NamE is KiShor','\$'
MOV AH,00H	.CODE
MOV AL,00H	MOV AX,@DATA
INT 10H;set video mode	MOV DS,AX
RET	MOV SI,OFFSET DATA
VIDEO_MODE ENDP	MOV CX,0017
CLEAR_SCREEN PROC	BACK:MOV AL,[SI]
MOV AH,06H	CMP AL,41H
·	JB OVER
MOV AL,00H	
MOV BH,07H	CMP AL,5AH
MOV CX,0000H	JA OVER
MOV DX,1827H	OR AL,00100000B;convert to lowercase
INT 10H; clear whole screen	OVER:MOV [SI],AL
RET	INC SI
CLEAR_SCREEN ENDP	LOOP BACK
SET_CURSOR PROC	CALL VIDEO_MODE
MOV AH,02H	CALL CLEAR_SCREEN
	CALL SET_CURSOR
MOV DH.12	5. III 52 55.155.1

MOV DH,12

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 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**





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

JA L1

CMP DX,000AH

ADD DX,30H

JMP L2	ADD DX,30H
L1:ADD DX,37H	PUSH DX
L2:PUSH DX	INC BX
INC BX	CMP AX,0000
CMP AX,0000	JA DCE
JA DCE	MOV CX,BX
MOV CX,BX	MOV AH,02
MOV AH,02	SHOW:POP DX
SHOW:POP DX	INT 21H
INT 21H	LOOP SHOW
LOOP SHOW	MOV AH,4C
RET	INT 21H
DISP ENDP	END
EXIT PROC	C:\MASM611\BIN>t9
MOV AH,4CH	9 518
INT 21H	TUTORIAL 10:
RET	.MODEL SMALL
EXIT ENDP	.STACK 100
END	.DATA
C:\MASM611\BIN>t8	.CODE
ZB8C6C	MOV AX,@DATA
TUTORIAL 09:	MOV DS,AX
.MODEL SMALL	CALL VIDEO_MODE
.STACK 64	CALL CLEAR_SCREEN
.DATA	CALL SET_CURSOR
SUM DW 0000	-
.CODE	CALL DISPLAY
MOV AX,@DATA	CALL EXIT_PROGRAM
MOV DS,AX	VIDEO_MODE PROC
MOV AH,01H	MOV AH,00H
INT 21H	MOV AL,00H
SUB AL,30H	INT 10H;set video mode
MOV CH,00H	RET
MOV CL,AL	VIDEO_MODE ENDP
MOV AH,02H	CLEAR_SCREEN PROC
MOV DL,20H	MOV AH,06H
INT 21H	MOV AL,00H
MOV AX,01	MOV BH,0BFH
MOV BX,03	MOV CX,0000H
ADD AX,BX	MOV DX,1827H
MOV SUM,AX	INT 10H; clear whole screen
•	RET
BACK:MOV DX,AX	CLEAR_SCREEN ENDP
MOV AX,BX	SET_CURSOR PROC
MOV BX,DX	MOV AH,02H
ADD SUM,BX	MOV DH,10
ADD AX,BX;addition of series	MOV DL,05
LOOP BACK	MOV BH,00
MOV AX,SUM;decimal display	INT 10H;set cursor at centre
MOV BX,0000	RET
MOV CX,0AH	SET_CURSOR ENDP
DCE:MOV DX,0000H	-
DIV CX	DISPLAY PROC; display characters MOV CX,5FH
	IVIO V CA.JI II

071-BEL-319 [KISHOR JOSHI]

	MOV BH,32	BACK:MOV HUN,00
LABEI	L1:MOV AH,02H	MOV TEN,00
	MOV DL,BH	MOV AH,00H
	INT 21H	MOV AL,[SI]
	INC BH	L1:CMP AX,64H
	LOOP LABEL1	JB NEXT1
	RET	INC HUN
	DISPLAY ENDP	SUB AX,64H
	EXIT_PROGRAM PROC	JMP L1
	MOV AH,4CH	NEXT1:CMP AX,0AH
	INT 21H	JB NEXT2
	RET	INC TEN
	EXIT_PROGRAM ENDP	SUB AX,0AH
	END	JMP NEXT1
C:\MA	NSM611\BIN>t10	NEXT2:MOV TEMP1,AX
		MOV AX,0001H
		MUL HUN
		MOV BX,100H
		MUL BX
		MOV TEMP,AX
	!"#\$z&' ()*+z0123456789;;<=>?@	MOV AX,0001H
CDEF	!"#\$x&'()*+,v0123456789:;<=>?@ GHIJKLMNOPQRSTUWXYZ[\]^_`abcdefgh	MUL TEN
	opgrstuvwxyz{}}~ ASM611\BIN>_	MOV BX,10H
		MUL BX
		ADD AX,TEMP
		ADD AX,TEMP1
		MOV [DI],AX;storing BCD value of data
<u>rutor</u>	IAL 11:	INC SI
MODE	L SMALL	INC DI
STACK	100	INC DI
DATA		DEC COUNT
	COUNT DB ?	JNZ BACK
	DATA1 DB 'kishor'	LEA SI,DATA1
	DATA2 DW 5 DUP(?)	MOV HUN,05
	SUM DW ?	BACK2:MOV AL,[SI] ;display BCD value of data
	TEMP DW ?	MOV AH,00
	TEMP1 DW ?	MOV BX,0000
	HUN DB 00	MOV CX,0AH
	TEN DB 00	DCE:MOV DX,0000H
CODE		DIV CX
	MOV AX,@DATA	ADD DX,30H
	MOV DS,AX	PUSH DX
	MOV AH,06H;clear screen	INC BX
	MOV AL,00H	CMP AX,0000
	MOV BH,07H	JA DCE
	MOV CX,0000H	MOV CX,BX
	MOV DX,1827H	MOV AH,02
	INT 10H;dos function	SHOW:POP DX
	LEA SI,DATA1	INT 21H
	LEA DI,DATA2; for storing BCD eqvt	LOOP SHOW
	MOV DX,0000H	MOV AH,02
	MOV COUNT,05H	MOV DI 20H

INT 21H	INC BX
INC SI	CMP AX,0000
DEC HUN	JA DCE
JNZ BACK2	MOV CX,BX
MOV AH,4CH	MOV AH,02
INT 21H	SHOW:POP DX
END	INT 21H
C:\MASM611\BIN>t11_	LOOP SHOW
	MOV AH,4CH
107 105 115 104 111	INT 21H
TUTORIAL 12:	END
.MODEL SMALL	C:\MASM611\BIN>t12
.STACK 64	kishor
.DATA	2
MSG DB 10 DUP(?)	TUTORIAL 13:
.CODE	.MODEL SMALL
MOV AX,@DATA	.STACK 64
MOV DS,AX	.DATA
MOV AH,0AH	MSG DB 10 DUP(?)
MOV CX,06H	.CODE
LEA DX,MSG	MOV AX,@DATA
INT 21H;string input	MOV DS,AX
MOV SI,OFFSET MSG	MOV AH,0AH
MOV CX,0006H	MOV CX,06H
MOV BX,0000H	LEA DX,MSG
ADD SI,02H	INT 21H;input string
BACK:MOV AH,[SI]	MOV SI,OFFSET MSG
CMP AH,61H;compairing vowel	MOV CX,0006H
JE NEXT	MOV BX,0000H
CMP AH,65H;compairing vowel	ADD SI,02H
JE NEXT	BACK:MOV AH,[SI]
CMP AH,69H;compairing vowel	CMP AH,61H;compairing vowel
JE NEXT	JE NEXT
CMP AH,6FH;compairing vowel	CMP AH,65H;compairing vowel
JE NEXT	JE NEXT
CMP AH,75H;compairing vowel	CMP AH,69H;compairing vowel
JE NEXT	JE NEXT
JMP GOTO1	CMP AH,6FH;compairing vowel
NEXT:INC BX;count vowel	JE NEXT
GOTO1:INC SI	CMP AH,75H;compairing vowel
LOOP BACK	JE NEXT
MOV AH,02	JMP GOTO1
MOV DL,OAH	NEXT:AND AH,11011111B;convert to uppercase
INT 21H;print space	MOV [SI],AH
;Display count in BCD	GOTO1:INC SI
MOV AX,BX	LOOP BACK
MOV BX,0000	MOV [SI],'\$'
MOV CX,0AH	CALL VIDEO_MODE
DCE:MOV DX,0000H	CALL CLEAR_SCREEN
DIV CX	CALL SET_CURSOR
ADD DX,30H	CALL DISPLAY
PUSH DX	CALL EXIT_PROGRAM

VIDEO_MODE PROC	INT 21H;string input
MOV AH,00H	MOV SI,OFFSET MSG
MOV AL,00H	ADD SI,02H
INT 10H;set video mode	MOV CX,0006H
RET	BACK:MOV AL,[SI]
VIDEO_MODE ENDP	CMP AL,61H
CLEAR_SCREEN PROC	JB OVER
MOV AH,06H	CMP AL,7AH
MOV AL,00H	JA OVER
MOV BH,07H	AND AL,11011111B; convert to uppercase
MOV CX,0000H	OVER:MOV [SI],AL
MOV DX,1827H	INC SI
INT 10H;clear screen	LOOP BACK
RET	MOV [SI],'\$'
CLEAR_SCREEN ENDP	CALL VIDEO_MODE
SET_CURSOR PROC	CALL CLEAR_SCREEN
MOV AH,02H	CALL SET_CURSOR
MOV DH,12	CALL DISPLAY
MOV DL,20	CALL EXIT_PROGRAM
MOV BH,00	VIDEO_MODE PROC
INT 10H;set cursor at centre	MOV AH,00H
RET	MOV AL,00H
SET_CURSOR ENDP	INT 10H;set video mode
DISPLAY PROC	RET
LEA DX,MSG	VIDEO_MODE ENDP
ADD DX,02H	CLEAR_SCREEN PROC
MOV AH,09H	MOV AH,06H
INT 21H; display string	MOV AL,00H
RET	MOV BH,07H
DISPLAY ENDP	MOV CX,0000H
EXIT_PROGRAM PROC	MOV DX,1827H
MOV AH,4CH	INT 10H;clear screen
INT 21H	RET
RET	CLEAR_SCREEN ENDP
EXIT_PROGRAM ENDP	SET_CURSOR PROC
END	MOV AH,02H
C:\MASM611\BIN>t13	MOV DH,12
aedgio_	MOV DL,20
	MOV BH,00
AEdg IO	INT 10H;set cursor at co ordinate RET
TUTORIAL 14:	SET_CURSOR ENDP
.MODEL SMALL	DISPLAY PROC
.STACK 64	LEA DX,MSG
.DATA	ADD DX,02H
MSG DB 10 DUP(?)	MOV AH,09H
.CODE	INT 21H; display string
MOV AX,@DATA	RET
MOV DS,AX	DISPLAY ENDP
MOV AH,0AH	EXIT_PROGRAM PROC
MOV CX,06H	MOV AH,4CH

INT 21H

LEA DX,MSG

```
RET
                                                         RET
      EXIT PROGRAM ENDP
                                                         CLEAR SCREEN ENDP
                                                         SET CURSOR PROC
      END
 C:\MASM611\BIN>t14
                                                         MOV AH,02H
kishor
                                                         MOV DH,10
                                                         MOV DL,02
                                                         MOV BH,00
                                                         INT 10H; set cursor at co ordinate
                        KISHOR
C:\MASM611\BIN>
                                                         RET
                                                         SET CURSOR ENDP
TUTORIAL 15:
                                                         DISPLAY PROC
.MODEL SMALL
                                                         LEA DX,MSG
.STACK 64
                                                         ADD DX,02H
.DATA
                                                         MOV AH,09H
      MSG DB 10 DUP(?)
                                                         INT 21H; display string
.CODE
                                                         RET
      MOV AX,@DATA
                                                         DISPLAY ENDP
      MOV DS,AX
                                                         EXIT_PROGRAM PROC
      MOV AH, OAH
                                                         MOV AH,4CH
      MOV CX,06H
                                                         INT 21H
      LEA DX,MSG
                                                         RET
      INT 21H;input string
                                                         EXIT PROGRAM ENDP
      MOV SI, OFFSET MSG
                                                         END
      ADD SI,02H
      MOV CX,0006H
                                                   C:\MASM611\BIN>t15
                                                   KISHOR
 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], '$'
                                                        kishor
      CALL VIDEO_MODE
                                                    C:\MASM611\BIN>
      CALL CLEAR SCREEN
                                                   TUTORIAL 16:
      CALL SET_CURSOR
                                                   .MODEL SMALL
      CALL DISPLAY
                                                   .STACK 200
      CALL EXIT PROGRAM
                                                   .DATA
      VIDEO MODE PROC
                                                         MSG DB 'KISHOR$'
      MOV AH,00H
                                                         TEMP1 DW 65535; for delay
      MOV AL,00H
                                                         TEMP2 DW 65535; for delay
      INT 10H;set video mode
                                                         TEMP3 DW 65535; for delay
      RET
                                                         TEMP4 DW 65535; for delay
      VIDEO_MODE ENDP
                                                         TEMP5 DW 65535; for delay
      CLEAR SCREEN PROC
                                                         LOC DW 0021H
      MOV AH,06H
                                                   .CODE
      MOV AL,00H
                                                         MOV AX,@DATA
      MOV BH,07H
                                                         MOV DS,AX
      MOV CX,0000H
                                                         CALL VIDEO_MODE
      MOV DX,1827H
                                                         MOV CX,25
```

INT 10H; clear screen

BACK:CALL CLEAR_SCREEN

MOV DX,LOC	DOCE and D.A. Conversable 2000 are as Transable 0.0 Progressy. T15.
CALL SET_CURSOR	Maria DOSBox 0.74. Cou speed: 3000 cycles. Frameskip 0. Program: 116
CALL DISPLAY	KISHOR
BACK3:DEC TEMP3;delay loop	
JNZ BACK3	TUTORIAL 17:
BACK1:DEC TEMP1;delay loop	.MODEL SMALL
JNZ BACK1	.STACK 64
BACK2:DEC TEMP2;delay loop	.DATA
JNZ BACK2	NUM DB 2 DUP(?)
BACK4:DEC TEMP4;delay loop	TEMP DW ?
JNZ BACK4	TEMP2 DW ?
BACK5:DEC TEMP5;delay loop	TEMP3 DW ?
JNZ BACK5	TEMP4 DW ?
SUB LOC,02	.CODE
LOOP BACK	MOV AX,@DATA
CALL EXIT_PROGRAM	MOV DS,AX
VIDEO_MODE PROC	MOV CX,0002
MOV AH,00H	LEA DI,NUM
MOV AL,00H	KJ:MOV AH,01H;input two digit
INT 10H;set video mode	INT 21H
RET	MOV [DI],AL
VIDEO_MODE ENDP	INC DI
CLEAR_SCREEN PROC	LOOP KJ
MOV AH,06H	MOV AH,02H
MOV AL,00H	MOV DL,20H
MOV BH,07H	INT 21H
MOV CX,0000H	MOV SI,OFFSET NUM
MOV DX,1827H	MOV DH,[SI]
INT 10H;clear screen	INC SI
RET	MOV DL,[SI]
CLEAR_SCREEN ENDP	SUB DH,30H;converting to eqvt HEX
SET_CURSOR PROC	SUB DL,30H
MOV AH,02H	MOV CL,04H
MOV BH,00	ROL DH,CL
INT 10H;set cursor	OR DH,DL
RET	MOV CL, DH
SET_CURSOR ENDP	MOV CH,00H
DISPLAY PROC	LABELS:CMP CL,10H
MOV AH,09H	JB NEXT
LEA DX,MSG	INC CH
INT 21H;display string	SUB CL,10H
RET	JMP LABELS
DISPLAY ENDP	NEXT:ADD CL,OAH
EXIT_PROGRAM PROC	DEC CH JNZ NEXT
MOV AH,4CH	MOV CH,00
INT 21H	•
RET	MOV TEMP,CX;HEX eqvt to temp
EXIT_PROGRAM ENDP	CALL CLEAR SCREEN
END	CALL CLEAR_SCREEN
DOSBox 0.74, Cou speed: 3000 cycles, Frameskip 0, Program: T16 -	MOV PL 01H
· · · · · · · · · · · · · · · · · · ·	MOV BL,01H
KISHOR_	MOV DH,00H

MOV DL,00H;set cursor at top at first	MOV BX,TEMP4
LABEL1:CALL SET_CURSOR	RET
MOV AX,TEMP	DECIMAL ENDP
MUL BL;multiplied at ax	END
CALL DECIMAL	12
INC BL	12 2 1
LOOP LABEL1	36
MOV AH,4CH	48
INT 21H	60 72
VIDEO MODE PROC	84
MOV AH,00H	96
MOV AL,00H	1 08 129
INT 10H;set video mode	128
RET	TUTORIAL 18:
VIDEO MODE ENDP	.MODEL SMALL
CLEAR SCREEN PROC	STACK 64
MOV AH,06H	.DATA
MOV AL,00H	NUM1 DW 0005
MOV BH,07H	NUM2 DW 0003
MOV CX,0000H	TEMP1 DW ?
MOV DX,1827H	
INT 10H;clear whole screen	TEMP2 DW ?
RET	SUM DW ?
CLEAR_SCREEN ENDP	.CODE
SET_CURSOR PROC	MOV AX,@DATA
MOV AH,02H	MOV DS,AX
MOV BH,00	MOV AX,NUM1
INC DH	MOV BX,NUM2
MOV DL,00	CMP AX,BX
INT 10H;set cursor	JA NEXT
RET	XCHG AX,BX;find greatest
SET CURSOR ENDP	MOV SUM,BX
_	NEXT:MOV DX,0000H;finding HCF
DECIMAL PROC; display eqvt BCD	MOV TEMP1,AX
MOV TEMP4,BX	MOV TEMP2,BX
MOV BX,0000H	DIV BX
MOV TEMP3,CX	CMP DX,0000H
MOV CX,000AH	JE LABEL1
MOV TEMP2,DX	DEC TEMP2
DCE: MOV DX,0000H	MOV AX,TEMP1
DIV CX	MOV BX,TEMP2
ADD DX,30H	JMP NEXT
PUSH DX	LABEL1:MOV AX,SUM
INC BX	MOV BX,TEMP2
CMP AX,0000	MOV DX,0000H
JA DCE	DIV BX
MOV CX,BX	CMP DX,0000H
MOV AH,02H	JE LABEL2
SHOW: POP DX	DEC TEMP2
INT 21H	MOV AX,TEMP1
LOOP SHOW	MOV BX,TEMP2
MOV DX,TEMP2	JMP NEXT
MOV CX,TEMP3	LAREL2·MOV AX TEMP2·HCE

	;BCD display	CMP DX,0000H	
	MOV BX,0000	JE LABEL2	
	MOV CX,0AH	DEC TEMP2	
DCE:	:MOV DX,0000H	MOV AX,TEMP1	
	DIV CX	MOV BX,TEMP2;HC	F
	ADD DX,30H	JMP NEXT	
	PUSH DX	LABEL2:MOV AX,NUM1	
	INC BX	MUL NUM2	
	CMP AX,0000	MOV DX,0000H	
	JA DCE	DIV TEMP2;LCM	
	MOV CX,BX	BCD display	
	MOV AH,02	MOV BX,0000	
SHO	W:POP DX	MOV CX,0AH	
	INT 21H	DCE:MOV DX,0000H	
	LOOP SHOW	DIV CX	
	MOV AH,4CH	ADD DX,30H	
	INT 21H	PUSH DX	
END		INC BX	
C:SMA	SM611\BIN>t18	CMP AX,0000	
5		JA DCE	
TUTOR	IAL 19:	MOV CX,BX	
.MODE	L SMALL	MOV AH,02	
.STACK	64	SHOW:POP DX	
.DATA		INT 21H	
	NUM1 DW 0005	LOOP SHOW	
	NUM2 DW 0003	MOV AH,4CH	
		•	
	TEMP1 DW ?	INT 21H	
	TEMP1 DW ? TEMP2 DW ?	INT 21H END	
.CODE	TEMP2 DW ?	END	
.CODE	TEMP2 DW ?	END C:\masm611\Bin>t19	
.CODE	TEMP2 DW ? SUM DW ?	END C:\MASM611\BIN>t19 15	
.CODE	TEMP2 DW ? SUM DW ? MOV AX,@DATA	END C:\MASM611\BIN>t19 15 TUTORIAL 20:	
.CODE	TEMP2 DW ? SUM DW ? MOV AX,@DATA MOV DS,AX	END C:\MASM611\BIN>t19 15 TUTORIAL 20: .MODEL SMALL	
.CODE	TEMP2 DW ? SUM DW ? MOV AX,@DATA MOV DS,AX MOV AX,NUM1	END C:\MASM611\BIN>t19 15 TUTORIAL 20: .MODEL SMALL .STACK 100	
.CODE	TEMP2 DW ? SUM DW ? MOV AX,@DATA MOV DS,AX MOV AX,NUM1 MOV BX,NUM2	END C:\MASM611\BIN>t19 15 TUTORIAL 20: .MODEL SMALL .STACK 100 .DATA	
.CODE	TEMP2 DW ? SUM DW ? MOV AX,@DATA MOV DS,AX MOV AX,NUM1 MOV BX,NUM2 CMP AX,BX	END C:\MASM611\BIN>t19 15 TUTORIAL 20: .MODEL SMALL .STACK 100 .DATA MSG DB 42 DUP(?)	
.CODE	TEMP2 DW ? SUM DW ? MOV AX,@DATA MOV DS,AX MOV AX,NUM1 MOV BX,NUM2 CMP AX,BX JA NEXT	END C:\MASM611\BIN>t19 15 TUTORIAL 20: .MODEL SMALL .STACK 100 .DATA MSG DB 42 DUP(?) TEMP DB ?	
	TEMP2 DW ? SUM DW ? MOV AX,@DATA MOV DS,AX MOV AX,NUM1 MOV BX,NUM2 CMP AX,BX JA NEXT XCHG AX,BX;find greatest	C:\MASM611\BIN>t19 15 TUTORIAL 20: .MODEL SMALL .STACK 100 .DATA MSG DB 42 DUP(?) TEMP DB ? .CODE	
	TEMP2 DW ? SUM DW ? 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	END C:\MASM611\BIN>t19 15 TUTORIAL 20: .MODEL SMALL .STACK 100 .DATA MSG DB 42 DUP(?) TEMP DB ? .CODE MOV AX,@DATA	
	TEMP2 DW ? SUM DW ? 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 MOV DX,0000H	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	
	TEMP2 DW ? SUM DW ? 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 MOV DX,0000H MOV TEMP1,AX	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	
	TEMP2 DW ? SUM DW ? 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 MOV DX,0000H MOV TEMP1,AX MOV TEMP2,BX	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	
	TEMP2 DW ? SUM DW ? 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 MOV DX,0000H MOV TEMP1,AX MOV TEMP2,BX DIV BX	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	
	TEMP2 DW ? SUM DW ? 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 MOV DX,0000H MOV TEMP1,AX MOV TEMP2,BX DIV BX CMP DX,0000H	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	
	TEMP2 DW ? SUM DW ? 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 MOV DX,0000H MOV TEMP1,AX MOV TEMP2,BX DIV BX CMP DX,0000H JE LABEL1	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	_
	TEMP2 DW ? SUM DW ? 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 MOV DX,0000H MOV TEMP1,AX MOV TEMP2,BX DIV BX CMP DX,0000H JE LABEL1 DEC TEMP2	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	_
	TEMP2 DW ? SUM DW ? 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 MOV DX,0000H MOV TEMP1,AX MOV TEMP2,BX DIV BX CMP DX,0000H JE LABEL1 DEC TEMP2 MOV AX,TEMP1	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 MS	_
NEXT:	TEMP2 DW ? SUM DW ? 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 MOV DX,0000H MOV TEMP1,AX MOV TEMP2,BX DIV BX CMP DX,0000H JE LABEL1 DEC TEMP2 MOV AX,TEMP1 MOV BX,TEMP2	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 MS ADD SI,02	G
NEXT:	TEMP2 DW ? SUM DW ? 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 MOV DX,0000H MOV TEMP1,AX MOV TEMP1,AX MOV TEMP2,BX DIV BX CMP DX,0000H JE LABEL1 DEC TEMP2 MOV AX,TEMP1 MOV BX,TEMP2 JMP NEXT	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 MS ADD SI,02 MOV [SI+0041],'\$'	G
NEXT:	TEMP2 DW ? SUM DW ? 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 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 1:MOV AX,SUM	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 MS ADD SI,02 MOV [SI+0041],'\$' CALL VIDEO_MODE	G

LABEL1:CALL SET_CURSOR;set cursor at every space	END
MOV AH,02H	C:\MASM611\BIN>t20
MOV TEMP,DL	programming in assembly language is fun_
BACK:MOV DL,[SI]	
CMP DL,20H;check space	🚟 DOSBox 0.74, Cbu speed: 3000 cycles, Frameski ນ
JNE NEXT	
INC DH	
MOV DL,TEMP	programming .
INC DL	in assembly
INC SI	language
JMP LABEL1	is
NEXT:CMP DL,'\$';check end point	- fun
JE GO1	TUTORIAL 21:
INC SI	.MODEL SMALL
INC TEMP	.STACK 64
INT 21H	.DATA
JMP BACK	NUM1 DB 1,2,3,4,5,6,7,8,9
GO1:CALL EXIT_PROGRAM	NUM2 DB 1,2,3,4,5,6,7,8,9
VIDEO_MODE PROC	NUM3 DB 9 DUP(?)
MOV AH,00H	TEMP1 DB 3
MOV AL,02H	TEMP2 DB 3
INT 10H;set video mode	.CODE
RET	MOV AX,@DATA
VIDEO_MODE ENDP	MOV DS,AX
CLEAR_SCREEN PROC	LEA SI,NUM1
MOV AH,06H	LEA BX,NUM2
MOV AL,00H	LEA DI,NUM3
MOV BH,07H	MOV CX,09
MOV CX,0000H	BACK:MOV AL,[SI]
MOV DX,1827H	ADD AL,[BX]
INT 10H;clear screen	MOV [DI],AL
RET	INC SI
CLEAR_SCREEN ENDP	INC DI
SET_CURSOR PROC	INC BX
MOV AH,02H	LOOP BACK
MOV BH,00	LEA SI,NUM3
INT 10H;set cursor as co ordinate	BACK2:MOV BX,0000
RET	MOV CX,000AH
SET_CURSOR ENDP	MOV AL,[SI]
EXIT_PROGRAM PROC	MOV AH,00
MOV AH,4CH	DDD: MOV DX,0000
INT 21H	DIV CX
RET	ADD DX,30H
EXIT_PROGRAM ENDP	PUSH DX
	INC BX
	CMP AX,0000
	JA DDD
	MOV CX,BX
	MOV AH,02
	SHOW:POP DX
	INT 21H

LOOP SHOW

INC SI	INC CH
MOV AH,02	SUB CL,10H
MOV DL,20H	JMP LABELS
INT 21H	NEXT:ADD CL,0AH;converted to eqvt HEX
DEC TEMP1	DEC CH
JNZ BACK2	JNZ NEXT
MOV TEMP1,03	BACK:ADD AX,CX
MOV AH,02	LOOP BACK
MOV DL,0AH	MOV SUM,AX;adding
INT 21H	MOV AX,SUM;BCD display
DEC TEMP2	MOV BX,0000
JNZ BACK2	MOV CX,0AH
MOV AH,4CH	DCE:MOV DX,0000H
INT 21H	DIV CX
END	ADD DX,30H
	PUSH DX
C:\MASM611\BIN>t21 2 4 6	INC BX
8 10 12	CMP AX,0000
14 16 18	JA DCE
TUTORIAL 22:	MOV CX,BX
.MODEL SMALL	MOV AH,02
.STACK 64	SHOW:POP DX
.DATA	INT 21H
SUM DW ?	LOOP SHOW
NUM DB 2 DUP(?)	MOV AH,02
.CODE	MOV DL,20H
MOV AX,@DATA	INT 21H;print space
MOV DS,AX	MOV AX,SUM;HEX display
MOV CX,0002	MOV BX,0000
LEA DI,NUM	MOV CX,10H
KJ:MOV AH,01H;input two digit no	
INT 21H	DCE1:MOV DX,0000H
MOV [DI],AL	DIV CX
INC DI	CMP DX,000AH
LOOP KJ	JA L3
MOV AH,02H	ADD DX,30H
MOV DL,20H	JMP L2
	L3:ADD DX,37H
INT 21H;print space	L2:PUSH DX
MOV SI,OFFSET NUM	INC BX
MOV DH,[SI]	CMP AX,0000
INC SI	JA DCE1
MOV DL,[SI]	MOV CX,BX
SUB DH,30H	MOV AH,02
SUB DL,30H	SHOW1:POP DX
MOV CL,04H	INT 21H
ROL DH,CL	LOOP SHOW1
OR DH,DL	MOV AH,4C
MOV AX,0000H	INT 21H
MOV CL,DH	END
MOV CH,00H	C:\MASM611\BIN>t22
LABELS:CMP CL,10H	15 120 78
JB NEXT	TUTORIAL 23:

071-BEL-319 [KISHOR JOSHI]

.MODEL SMALL	MOV SUM,AX
STACK 64	MOV CH,00H
.DATA	MOV AX,SUM;decimal;display;of;sum
SUM DW ?	MOV BX,0000
NUM DB 2 DUP(?)	MOV CX,0AH
.CODE	DCE:MOV DX,0000H
MOV AX,@DATA	DIV CX
MOV DS,AX	ADD DX,30H
MOV CX,0002	PUSH DX
LEA DI,NUM	INC BX
KJ:MOV AH,01H;input 2 digit no	CMP AX,0000
INT 21H	JA DCE
MOV [DI],AL	MOV CX,BX
INC DI	MOV AH,02
LOOP KJ	SHOW:POP DX
MOV AH,02H	INT 21H
MOV DL,20H	LOOP SHOW
INT 21H	MOV AH,02
MOV SI,OFFSET NUM	MOV DL,20H
MOV DH,[SI]	INT 21H
INC SI	MOV AX,SUM;HEX;display
MOV DL,[SI]	MOV BX,0000
SUB DH,30H	MOV CX,10H
SUB DL,30H	DCE1:MOV DX,0000H
MOV CL,04H	DIV CX
ROL DH,CL	CMP DX,000AH
OR DH,DL	JA L3
MOV CL DH	ADD DX,30H JMP L2
MOV CL OOL	
MOV CH,00H	L3:ADD DX,37H
LABELS:CMP CL,10H JB NEXT	L2:PUSH DX
	INC BX
INC CH	CMP AX,0000
SUB CL,10H	JA DCE1
JMP LABELS	MOV CX,BX
NEXT:CMP CH,00H	MOV AH,02
JA KJJ	SHOW1:POP DX
JMP JK	INT 21H
KJJ:ADD CL,0AH;converted to eqvt HEX	LOOP SHOW1
DEC CH	MOV AH,4C
JNZ KJJ;INPUT	INT 21H
JK:MOV BX,0000H	END
BACK:MOV AX,CX	C:\MASM611\BIN>t23
INC AX	96 278 4E
MOV DL,AL	TUTORIAL 24:
MUL DL	.MODEL SMALL
ADD BX,AX;addition of series	.STACK 100
LOOP BACK	.DATA
MOV DX,0000H	TRUE DB 'THE NUMBER IS PRIME\$'
MOV AX,BX	FALSE DB 'THE NUMBER IS NOT PRIME\$'
MOV CX,0002	SUM DW ?
MUL CX	NUM DB 2 DUP(?)

TEMP DW ?	JMP EN
CODE	NO:LEA DX,TRUE
MOV AX,@DATA	MOV AH,09H
MOV DS,AX	INT 21H
MOV CX,0002	EN:MOV AH,4CH
LEA DI,NUM	INT 21H
KJ:MOV AH,01H	END
INT 21H	C:\MASM611\BIN>t24
MOV [DI],AL	97 THE NUMBER IS PRIME
INC DI	C:\MASM611\BIN>t24
LOOP KJ	88 THE NUMBER IS NOT PRIME
MOV AH,02H	TUTORIAL 25:
MOV DL,20H	.MODEL SMALL
INT 21H	.STACK 100H
MOV SI,OFFSET NUM	.DATA
MOV DH,[SI]	MSG DB "TODAY'S DATE IS ",'\$'
INC SI	MSG1 DB "TODAY'S TIME IS ",'\$'
MOV DL,[SI]	.CODE
SUB DH,30H	MOV AX,@DATA
SUB DL,30H	MOV DS,AX
MOV CL,04H	MOV DX,OFFSET MSG
ROL DH,CL	MOV AH,09H
OR DH,DL	INT 21H
MOV AX,0000H	MOV AH,2AH
MOV CL,DH	INT 21H
MOV CH,00H	PUSH CX
LABELS:CMP CL,10H	MOV CX,0
JB NEXT	MOV CL,DL
INC CH	PUSH CX
SUB CL,10H	MOV CL,DH
JMP LABELS	PUSH CX
NEXT:CMP CH,00H	MOV DH,0
JA KJJ	MOV DX,0
JMP JK	POP AX
KJJ:ADD CL,OAH	MOV CX,0
DEC CH	MOV BX,10
JNZ KJJ;INPUT	DIVIDE1: DIV BX
JK:MOV AX,CX	PUSH DX
MOV TEMP,AX	ADD CX,1
DEC CX	MOV DX,0
DEC CX	CMP AX,0
BACKA:MOV DX,CX	JNE DIVIDE1
INC DX	DIVIDE2: POP DX
	ADD DL,30H
DIV DL	MOV AH,02H
CMP AH,00H	INT 21H
JE YES	LOOP DIVIDE2
MOV AX,TEMP	MOV DL,'/'
LOOP BACKA	MOV AH,02H
JMP NO	INT 21H
YES:LEA DX,FALSE	MOV DX,0
MOV AH,09H	POP AX
INT 21H	

PUSH DX MOV CX,0 MOV BX,10 ADD CX,1 **DIVIDE3: DIV BX** MOV DX,0 **PUSH DX** CMP AX,0 ADD CX,1 JNE TDIVIDE1 MOV DX,0 TDIVIDE2: POP DX CMP AX,0 ADD DL,30H JNE DIVIDE3 MOV AH,02H **DIVIDE4: POP DX INT 21H** ADD DL,30H LOOP TDIVIDE2 MOV DL,'-' MOV AH,02H **INT 21H** MOV AH,02H LOOP DIVIDE4 **INT 21H** MOV DL,'/' MOV DX,0 POP AX MOV AH,02H INT 21H MOV CX,0 MOV DX,0 MOV BX,10 POP AX TDIVIDE3: DIV BX MOV CX,0 **PUSH DX** MOV BX,10 ADD CX,1 **DIVIDE5: DIV BX** MOV DX,0 **PUSH DX** CMP AX.0 ADD CX,1 JNE TDIVIDE3 MOV DX,0 TDIVIDE4: POP DX CMP AX,0 ADD DL,30H JNE DIVIDE5 MOV AH,02H **DIVIDE6: POP DX INT 21H** ADD DL,30H LOOP TDIVIDE4 MOV AH,02H MOV DL,'-' **INT 21H** MOV AH,02H INT 21H LOOP DIVIDE6 MOV DL,0AH MOV DX,0 MOV AH,02H POP AX **INT 21H** MOV CX.0 MOV DX, OFFSET MSG1 MOV BX,10 MOV AH,09H TDIVIDE5: DIV BX **PUSH DX INT 21H** MOV AH,2CH ADD CX,1 INT 21H MOV DX,0 MOV DL, DH CMP AX.0 MOV DH,0 JNE TDIVIDE5 **PUSH DX** TDIVIDE6: POP DX MOV DL,CL ADD DL,30H **PUSH DX** MOV AH,02H MOV DL,CH **INT 21H PUSH DX** LOOP TDIVIDE6 MOV DH,0 MOV AH,4CH MOV DX,0 **INT 21H** POP AX **END** MOV CX,0 C:\MASM611\BIN>t25

MOV BX,10

TDIVIDE1: DIV BX

TODAY'S DATE IS 7/30/2016 TODAY'S TIME IS 14-17-12