CASE CONVERSION

ASSEMBLED CODE	DISASSEMBLED CODE
ASSUME CS: CODE, DS: DATA	
; ---	
DATA SEGMENT	
COUNT EQU 10H	
DATA ENDS	
; ---	
CODE SEGMENT	
START: MOV AX, DATA	MOV AX,076A
MOV DS, AX	MOV DS, AX
MOV CX, COUNT	MOV CX, 0010
L1: MOV AH, 1	MOV AH, 01
INT 21H	INT 21H
; AL = CHARACTER	
;ASCII (HEX): A-Z=41-5A, a-z=61-7A	
CMP AL,60H	CMP AL,6-H
JNC UPPER	JNC 0014
ADD AL,20H	ADD AL, 20
JMP SKIP	JMP 0016
UPPER: SUB AL,20H	SUB AL, 20
; CONVERT TO UPPER CASE	
SKIP: MOV AH,2	MOV AH, 02
MOV DL, AL; CHARACTER MUST BE IN DL	
INT 21H; DISPLAY THE CHARACTER	
LOOP L1; REPEAT LOOP	MOV DL, AL
MOV AH,4CH	INT 21H
INT 21H	LOOP 0008
CODE ENDS	MOV AH, 4C
END START	INT 21H

```
D:\>masm EX8.ASM
Microsoft (R) MASM Compatibility Driver
Copyright (C) Microsoft Corp 1993.  All rights reserved.

 Invoking: ML.EXE /I. /Zm /c /Ta EX8.ASM

Microsoft (R) Macro Assembler Version 6.11
Copyright (C) Microsoft Corp 1981-1993.  All rights reserved.

 Assembling: EX8.ASM

D:\>link EX8.OBJ,,;

   Microsoft Object Linker V2.01 (Large)
(C) Copyright 1982, 1983 by Microsoft Inc.

Warning: No STACK segment

There was 1 error detected.

D:\>EX8.EXE
aAbBcCdDeEfFGgHhIiJjKkLlMmNnOoPp
D:\>_
```

FLOATING POINT ADDITION

ASSEMBLED CODE	DISASSEMBLED CODE
ASSUME CS: CODESEG, DS: DATASEG	
; ----------	
DATASEG SEGMENT	
ORG 00H	
X DD 20.4375	
ORG 10H	
Y DD 20.4375	
ORG 20H	
SUM DD ?	
DATASEG ENDS	
; ----------	
CODESEG SEGMENT	
START: MOV AX, DATASEG	MOV AX,076A
MOV DS, AX; ASSIGN VALUE TO DS	MOV DS, AX
FINIT; INITIALIZE STACK	FINIT
FLD X; LOAD X INTO ST (0)	FLD DWORD PTR[0000]
FLD Y; LOAD Y INTO ST (0)	FLD DWORD PTR[0010]
FADD ST (0), ST (1); ST (0) = X+Y	FADD ST, ST (1)
FST SUM; STORE ST (0) IN SUM	FST DWORD PTR[0020]
MOV AH,4CH	MOV AH, 4C
INT 21H	INT 21H
CODESEG ENDS	
END START	

```
-d 076A:0000
076A:0000  00 80 A3 41 00 00 00 00-00 00 00 00 00 00 00 00   ...A............
076A:0010  00 80 A3 41 00 00 00 00-00 00 00 00 00 00 00 00   ...A............
076A:0020  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
076A:0030  B8 6A 07 8E D8 9B DB E3-9B D9 06 00 00 9B D9 06   .j..............
076A:0040  10 00 9B D8 C1 9B D9 16-20 00 B4 4C CD 21 F8 B7   ........ ..L.!..
076A:0050  00 8A 87 48 2F D0 D8 73-17 E8 B6 00 8A 5E F8 B7   ...H/..s.....^..
076A:0060  00 8A 87 48 2F D0 D8 73-07 53 B0 01 50 E8 73 01   ...H/..s.S..P.s.
076A:0070  A0 B6 2C 3A 46 F8 74 7E-C7 46 FA 00 00 8A 46 F8   ..,:F.t~.F....F.
-g

Program terminated normally
-d 076A:0000
076A:0000  00 80 A3 41 00 00 00 00-00 00 00 00 00 00 00 00   ...A............
076A:0010  00 80 A3 41 00 00 00 00-00 00 00 00 00 00 00 00   ...A............
076A:0020  00 80 23 42 00 00 00 00-00 00 00 00 00 00 00 00   ..#B............
076A:0030  B8 6A 07 8E D8 9B DB E3-9B D9 06 00 00 9B D9 06   .j..............
076A:0040  10 00 9B D8 C1 9B D9 16-20 00 B4 4C CD 21 F8 B7   ........ ..L.!..
076A:0050  00 8A 87 48 2F D0 D8 73-17 E8 B6 00 8A 5E F8 B7   ...H/..s.....^..
076A:0060  00 8A 87 48 2F D0 D8 73-07 53 B0 01 50 E8 73 01   ...H/..s.S..P.s.
076A:0070  A0 B6 2C 3A 46 F8 74 7E-C7 46 FA 00 00 8A 46 F8   ..,:F.t~.F....F.
-q

D:\>_
```

FLOATING POINT SUBTRACTION

ASSEMBLED CODE	DISASSEMBLED CODE
ASSUME CS: CODESEG, DS: DATASEG	
; ----------	
DATASEG SEGMENT	
ORG 00H	
X DD 20.4375	
ORG 10H	
Y DD 20.4375	
ORG 20H	
DIFF DD ?	
DATASEG ENDS	
; ----------	
CODESEG SEGMENT	
START: MOV AX, DATASEG	MOV AX,076A
MOV DS, AX; ASSIGN VALUE TO DS	MOV DS, AX
FINIT; INITIALIZE STACK	FINIT
FLD X; LOAD X INTO ST (0)	FLD DWORD PTR[0000]
FLD Y; LOAD Y INTO ST (0)	FLD DWORD PTR[0010]
FSUB ST (0), ST (1); ST (0) = X+Y	FSUB ST, ST (1)
FST DIFF; STORE ST (0) IN DIFF	FST DWORD PTR[0020]
MOV AH,4CH	MOV AH, 4C
INT 21H	INT 21H
CODESEG ENDS	
END START	

```
-d 076A:0000
076A:0000  00 80 A3 41 00 00 00 00-00 00 00 00 00 00 00 00   ...A............
076A:0010  00 00 00 3E 00 00 00 00-00 00 00 00 00 00 00 00   ...>............
076A:0020  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
076A:0030  B8 6A 07 8E D8 9B DB E3-9B D9 06 10 00 9B D9 06   .j..............
076A:0040  00 00 9B D8 E1 9B D9 16-20 00 B4 4C CD 21 F8 B7   ........ ..L.!..
076A:0050  00 8A 87 48 2F D0 D8 73-17 E8 B6 00 8A 5E F8 B7   ...H/..s.....^..
076A:0060  00 8A 87 48 2F D0 D8 73-07 53 B0 01 50 E8 73 01   ...H/..s.S..P.s.
076A:0070  A0 B6 2C 3A 46 F8 74 7E-C7 46 FA 00 00 8A 46 F8   ..,:F.t~.F....F.
-g

Program terminated normally
-d 076A:0000
076A:0000  00 80 A3 41 00 00 00 00-00 00 00 00 00 00 00 00   ...A............
076A:0010  00 00 00 3E 00 00 00 00-00 00 00 00 00 00 00 00   ...>............
076A:0020  00 80 A2 41 00 00 00 00-00 00 00 00 00 00 00 00   ...A............
076A:0030  B8 6A 07 8E D8 9B DB E3-9B D9 06 10 00 9B D9 06   .j..............
076A:0040  00 00 9B D8 E1 9B D9 16-20 00 B4 4C CD 21 F8 B7   ........ ..L.!..
076A:0050  00 8A 87 48 2F D0 D8 73-17 E8 B6 00 8A 5E F8 B7   ...H/..s.....^..
076A:0060  00 8A 87 48 2F D0 D8 73-07 53 B0 01 50 E8 73 01   ...H/..s.S..P.s.
076A:0070  A0 B6 2C 3A 46 F8 74 7E-C7 46 FA 00 00 8A 46 F8   ..,:F.t~.F....F.
-q

D:\>
```

DISPLAY A STRING

ASSEMBLED CODE	DISASSEMBLED CODE
ASSUME CS: CODE, DS:DATA	
; ---	
DATA SEGMENT	
MESSAGE DB "THIS IS THE STRING$"	
DATA ENDS	
; ---	
CODE SEGMENT	
START: MOV AX, DATA	MOV AX,076A
MOV DS, AX	MOV DS, AX
MOV AH,9; DOS FUNCTION #9	MOV AH, 09
MOV DX, OFFSET MESSAGE	MOV DX, 0000
; OFFSET OF THE STRING	INT 21H
INT 21H; DISPLAY IT	MOV AH, 4C
MOV AH,4CH	INT 21H
INT 21H	
CODE ENDS	
END START	

```
D:\>masm EX10.ASM
Microsoft (R) MASM Compatibility Driver
Copyright (C) Microsoft Corp 1993.  All rights reserved.

 Invoking: ML.EXE /I. /Zm /c /Ta EX10.ASM

Microsoft (R) Macro Assembler Version 6.11
Copyright (C) Microsoft Corp 1981-1993.  All rights reserved.

 Assembling: EX10.ASM

D:\>link EX10.OBJ,,;

   Microsoft Object Linker V2.01 (Large)
(C) Copyright 1982, 1983 by Microsoft Inc.

Warning: No STACK segment

There was 1 error detected.

D:\>EX10.EXE
THIS IS THE STRING
D:\>
```

SYSTEM DATE

ASSEMBLED CODE	DISASSEMBLED CODE
ASSUME CS: CODE, DS: DATA	
; ----------	
DATA SEGMENT	
DAY DB 01 DUP(?)	
MONTH DB 01 DUP(?)	
YEAR DB 02 DUP(?)	
DATA ENDS	
; ----------	
CODE SEGMENT	
ORG 0100H	
START: MOV AX, DATA	MOV AX, 076A
MOV DS, AX	MOV DS, AX
; INT 21H /AH=2AH - GET SYSTEM DATE.	
; RETURN: CX= YEAR (1980-2099). DH= MONTH.	
; DL= DAY.AL= DAY OF WEEK (00H=SUNDAY)	
MOV AH,2AH	MOV AH, 2A
INT 21H	INT 21H
MOV SI, OFFSET DAY	MOV SI, 0000
MOV [SI], DL	MOV [SI], DL
MOV SI,OFFSET MONTH	MOV SI, 0001
MOV [SI], DH	MOV [SI], DH
MOV SI,OFFSET YEAR	MOV SI, 0002
MOV [SI], CX	MOV [SI], CX
MOV AH,4CH	MOV AH, 4C
INT 21H	INT 21H
CODE ENDS	
END START	

```
-d 076a:0000
076A:0000  0B 0A E6 07 00 00 00 00-00 00 00 00 00 00 00 00   ................
076A:0010  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
076A:0020  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
076A:0030  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
076A:0040  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
076A:0050  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
076A:0060  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
076A:0070  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
```

SYSTEM TIME

ASSEMBLED CODE	DISASSEMBLED CODE
ASSUME CS: CODE, DS: DATA	
; ----------	
DATA SEGMENT	
HOUR DB 01 DUP(?)	
MINUTE DB 01 DUP(?)	
SECOND DB 02 DUP(?)	
DATA ENDS	
; ----------	
CODE SEGMENT	
ORG 0100H	
START: MOV AX, DATA	MOV AX, 076A
MOV DS, AX	MOV DS, AX
; INT 21H/AH=2CH- GET SYSTEM TIME.	
; RETURN: CH= HOUR. CL= MIN. DH= SEC	
MOV AH,2CH	MOV AH, 2C
INT 21H	INT 21H
MOV SI, OFFSET HOUR	MOV SI, 0000
MOV [SI], CH	MOV [SI], CH
MOV SI, OFFSET MINUTE	MOV SI, 0001
MOV [SI], CL	MOV [SI], CL
MOV SI, OFFSET SECOND	MOV SI, 0002
MOV [SI], DH	MOV [SI], DH
MOV AH,4CH	MOV AH, 4C
INT 21H	INT 21H
CODE ENDS	
END START