UCS1512 - MICROPROCESSOR S LAB

CASE CONVERSION

Exp No: 8

Name : Anirudh

Reg No: 185001019

Aim

To write a program for performing case conversion on the fly in an 8086 microprocessor using MASM and DOSBox.

Algorithm

1. Define the values in the data segment and assign the counter value
2. Initialize the data segment register with a data segment address
3. Load the loop counter value into CX which determines how many inputs for case conversion can be given by the user before the program terminates
4. Repeat till CX becomes zero:
    4.1. Call the DOS Interrupt Function 21H with AH = 01H to read a single character keyboard input with echo into AL whose value will lie in the range:
        - ASCII(hex) : A - Z = 41 - 5A, a - z = 61 - 7A
        - ASCII(dec) : A - Z = 65 - 90, a - z = 97 - 122
    4.2. Check if the character was uppercase or lowercase using CMP:
        - Uppercase: < 60H which is (dec)96
        - Lowercase: > 60H which is (dec)96
    4.3. If the character read was lower case, go to step 4.5
    4.4. If the character read was uppercase, convert to lowercase by adding 20H which is 32(dec):
        - A: 41H a:61H

- a - A = 20HIf the character read was lowercase, convert to uppercase by subtracting 20H which is 32(dec)

4.5. Display the converted character by calling the DOS Interrupt Function 21H with AH = 02H to display a single character stored in DL

5. Terminate the program by calling the DOS Interrupt Function 21H with AH = 4CH

Program	Comments
ASSUME CS:CODE,DS:data	Naming the CS and DS for the program
data SEGMENT COUNT EQU 10h data ends	When EQU refers to a constant value, it becomes a synonym for that value and this value can't be overwritten
CODE SEGMENT org 0100h	Providing an offset value

START:
MOV AX,data
MOV DS,AX
	Initializing the data segment register with the data segment address

MOV CX,COUNT	Loading the count into CX for looping
L1: MOV AH,1 INT 21H	Call the DOS Interrupt Function 21H with AH = 01H to read a single character keyboard input with echo into AL

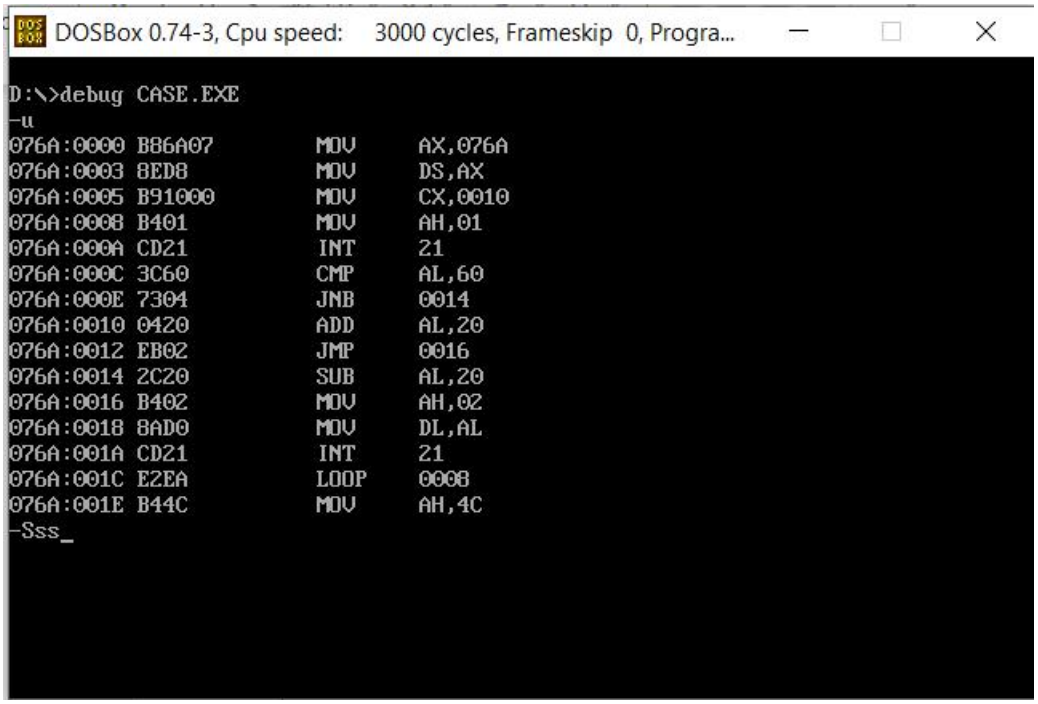
CMP AL,60H
JNC UPPER
ADD AL,20H
	Check if the character was uppercase or lowercase using CMP and jump if it was lowercase. Otherwise, convert to lowercase.

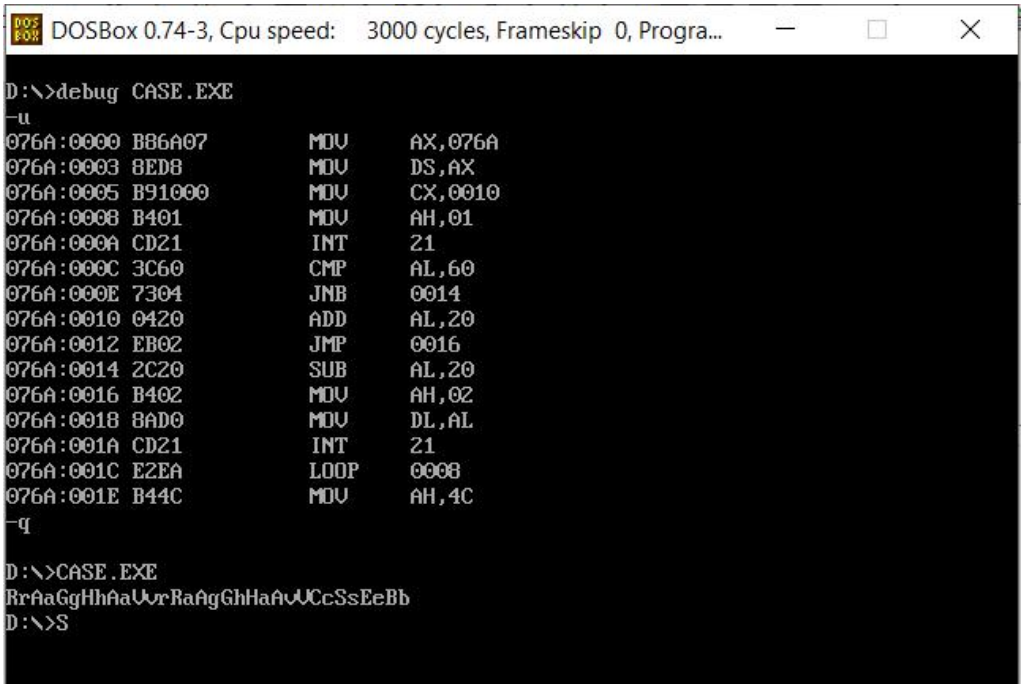
JMP SKIP	Skip to display if input was uppercase
UPPER: SUB AL,20H	Convert to uppercase when lowercase input is detected

SKIP:
MOV AH,2
MOV DL,AL
INT 21H
LOOP L1
	Display the converted character by calling the DOS Interrupt Function 21H with AH = 02H to display a single character stored in DL

MOV Ah,4CH INT 21H CODE ENDS end start	Calling the DOS Interrupt Function 21H with AH = 4CH to terminate the program

Snapshot





Result

Program for performing case conversion on the fly in an 8086 microprocessor using MASM and DOSBox was implemented and the output was verified.

FLOATING POINT OPERATIONS

Exp No: 9

Name : Anirudh H

Reg No: 185001019

Aim

To write programs for performing floating point operations in an 8086 microprocessor using MASM and DOSBox.

Algorithm

6. Define the values in the data segment and assign the initial values if required
7. Initialize the data segment register with a data segment address
8. Initialize the 8087 stack
9. Load the data into the 8087 stack
10. Perform addition/ subtraction and store the sum/ difference to the result address in 8086 for display
11. Terminate the program

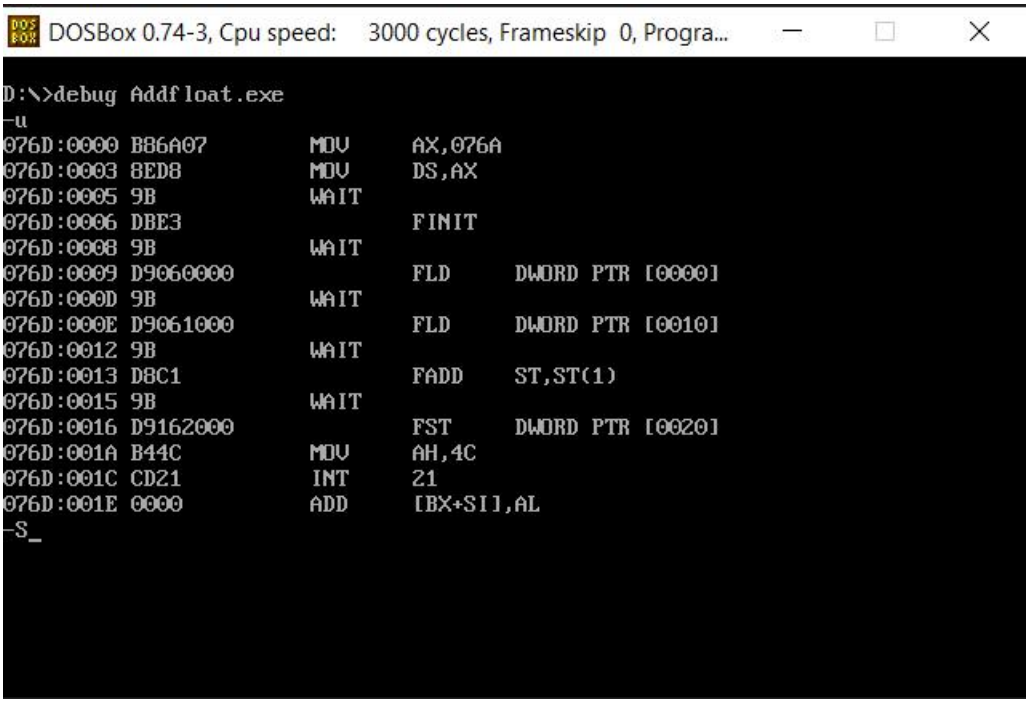
Program for adding two floating point numbers

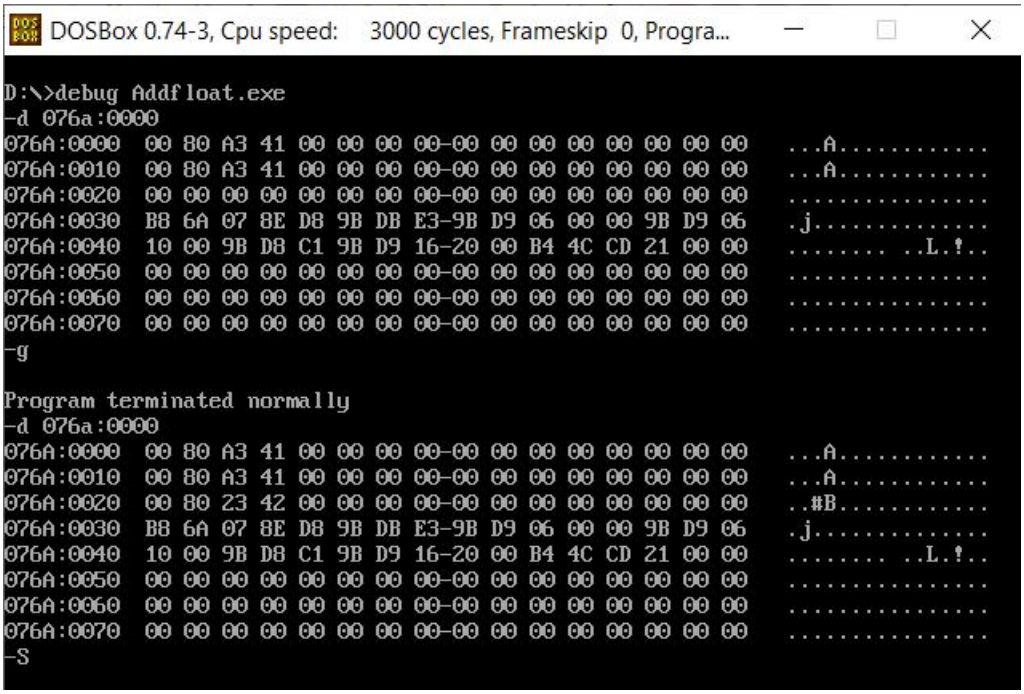
Program	Comments

CODESEG SEGMENT
start:
MOV AX,DATASEG
MOV DS,AX
	Initializing the data segment register with the data segment address

FINIT	Initialize the 8087 stack
FLD X FLD Y	Load X into ST(0) Load Y into ST(0), X is now in ST(1)
FADD ST(0),ST(1)	Performing addition: ST(0) = Y[ST(0)] + X[ST(1)]
FST SUM	Store ST(0) in sum
MOV AH,4CH INT 21H CODESEG ENDS END START	Calling the DOS Function to enter the display screen using interrupt 21H and to terminate the program

Snapshot:


DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Progra...
D:\>debug Addfloat.exe
-u
076D:0000 B86A07 MOV AX,076A
076D:0003 8ED8 MOV DS,AX
076D:0005 9B WAIT
076D:0006 DBE3 FINIT
076D:0008 9B WAIT
076D:0009 D9060000 FLD DWORD PTR [0000]
076D:000D 9B WAIT
076D:000E D9061000 FLD DWORD PTR [0010]
076D:0012 9B WAIT
076D:0013 D8C1 FADD ST,ST(1)
076D:0015 9B WAIT
076D:0016 D9162000 FST DWORD PTR [0020]
076D:001A B44C MOV AH,4C
076D:001C CD21 INT 21
076D:001E 0000 ADD [BX+SI],AL
-S_


DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Progra...
D:\>debug Addfloat.exe
-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 00 00 ....L.!..
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 ....
-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 00 00 ....L.!..
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 ....
-S

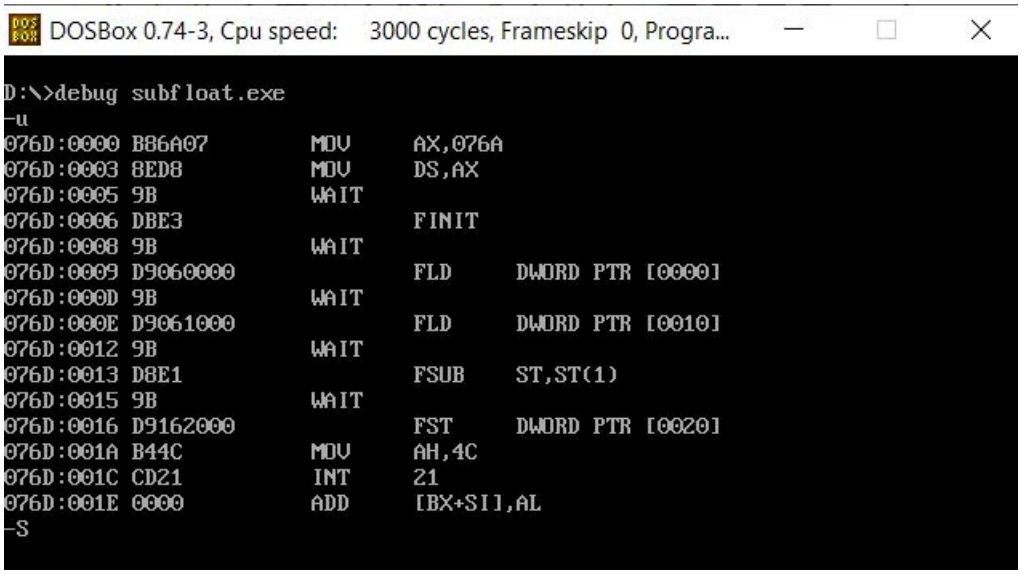
Program for subtracting two floating point numbers

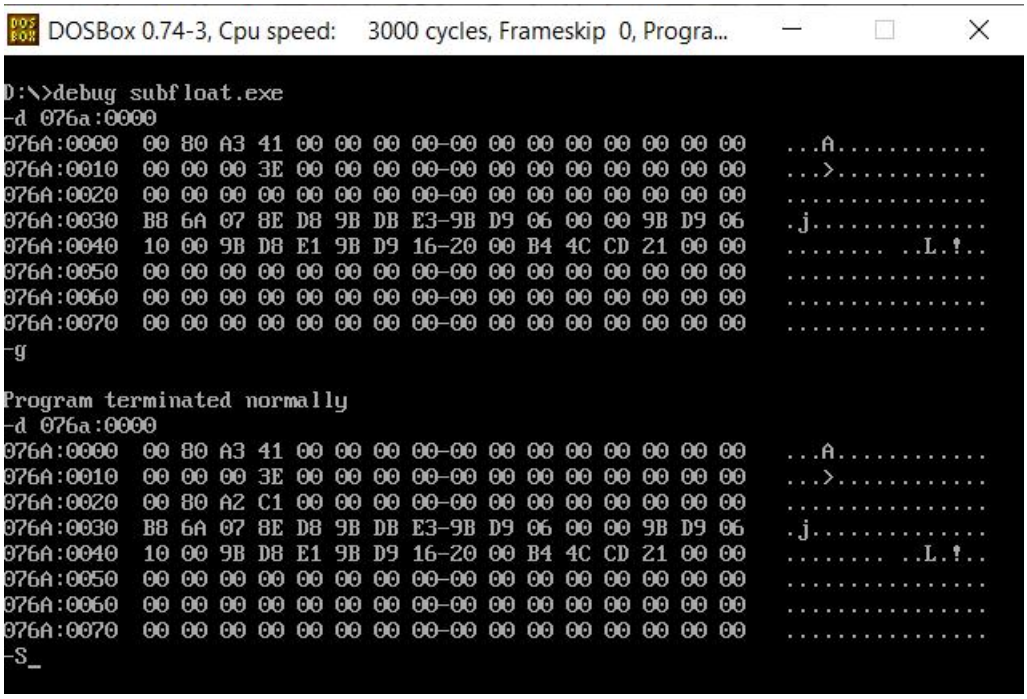
Program	Comments

CODESEG SEGMENT
start:
MOV AX,DATASEG
MOV DS,AX
	Initializing the data segment register with the data segment address

FINIT	Initialize the 8087 stack
FLD Y FLD X	Load Y into ST(0) Load X into ST(0), Y is now in ST(1)
FSUB ST(0),ST(1)	Performing subtraction: ST(0) = X[ST(0)] - Y[ST(1)]
FST DIFF	Store ST(0) in diff
MOV AH,4CH INT 21H CODESEG ENDS END START	Calling the DOS Function to enter the display screen using interrupt 21H and to terminate the program

Snapshot:





Result

Programs for performing floating point operations in an 8086 microprocessor using MASM and DOSBox were implemented and the outputs were verified.