BCD Addition and Subtraction

Expt No:	7	**Name:**	Mahesh Bharadwaj K
Date :	08/10/2020	**Reg No:**	185001089

Aim:

To write and execute 8086 programs for BCD addition and subtraction.

Procedure:

- Mount masm folder to a drive on DOSBOX.
- Navigate to mounted drive using 'dir' .
- Save 8086 program with the extension **'.asm'** in the same folder using the command **'edit'**.
- Assemble the **.asm** file using the command **'masm filename.asm'**.
- Link the assmebled **.obj** file using the command **'link filename.obj'**.
- Debug the executable file **.exe** with the **'debug filename.exe'** command.
    i. **U:** To view the un-assembled code.
    ii. **D:** Used as 'D segment:offset' to see the content of memory locations starting from segment:offset address.
    iii. **E:** To change the values in memory.
    iv. **G:** Execute the program using command.
    v. **Q** exits from the debug session.

Algorithm:

1.BCD Addition

* First operand is stored in opr1 & second operand is stored in opr2.
* Move the data segment address to the AX register and then move it to the DS register.
* Move opr1 into AL using MOV.
* Move opr2 into BL using MOV.
* Add AL, BL using ADD.
* Decimal adjust after addition using DAA.
* IF there is a carry, move 01 into carry variable.

* ELSE move 00 into carry variable.
* Move value in AL into result using MOV.

2. BCD Subtraction

* First operand is stored in opr1 & second operand is stored in opr2.
* Move the data segment address to the AX register and then move it to the DS register.
* Move opr1 into AL using MOV.
* Move opr2 into BL using MOV.
* Subtract BL from AL using SUB.
* Decimal adjust after subtraction using DAS.
* IF borrow(carry) flag is NOT set, move 00 into sign variable, and AL into difference using MOV.
* ELSE,
  - Move 01h into sign variable.
  - Mov 99 into BL register.
  - Subtract AL from BL using SUB.
  - Increment BL by 1.
  - BL now contains 10s complement of AL.
  - Move BL into difference.

---

1. BCD Addition

Program:

Program		Comments
start:	MOV AX,data	Move data segment address contents to AX register
	MOV ds,AX	Move data in AX register to DS register
	MOV AL, opr1	Move first operand into AL
	mov BL ,opr2	Move second operand into BL
	ADD AL, BL	Add AL and BL
	DAA	Decimal Adjust after addition
	JNC **skip**	IF there is no carry skip
	MOV carry, 01h	Load 01 into carry
skip:	MOV sum,AL	Store sum obtained from AL register
	MOV ah,4ch	
	INT 21h	Request interrupt routine

Unassembled Code:

```
D:\>debug 7A.EXE
-U
076C:0100 B86A07        MOV     AX,076A
076C:0103 8ED8          MOV     DS,AX
076C:0105 A00000        MOV     AL,[0000]
076C:0108 8A1E0100      MOV     BL,[0001]
076C:010C 02C3          ADD     AL,BL
076C:010E 27            DAA
076C:010F 7305          JNB     0116
076C:0111 C606110001    MOV     BYTE PTR [0011],01
076C:0116 A21000        MOV     [0010],AL
076C:0119 B44C          MOV     AH,4C
076C:011B CD21          INT     21
```

Input and Output:

Figure 1: **Input:** opr_1 = 32 & opr_2 = 86 **Output:** carry: 01, sum: 18

```
-d 076A:0000
076A:0000  32 86 00 00 00 00 00 00-00 00 00 00 00 00 00 00   2...............
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   ................
-g

Program terminated normally
-d 076A:0000
076A:0000  32 86 00 00 00 00 00 00-00 00 00 00 00 00 00 00   2...............
076A:0010  18 01 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   ................
```

2. BCD Subtraction

Program:

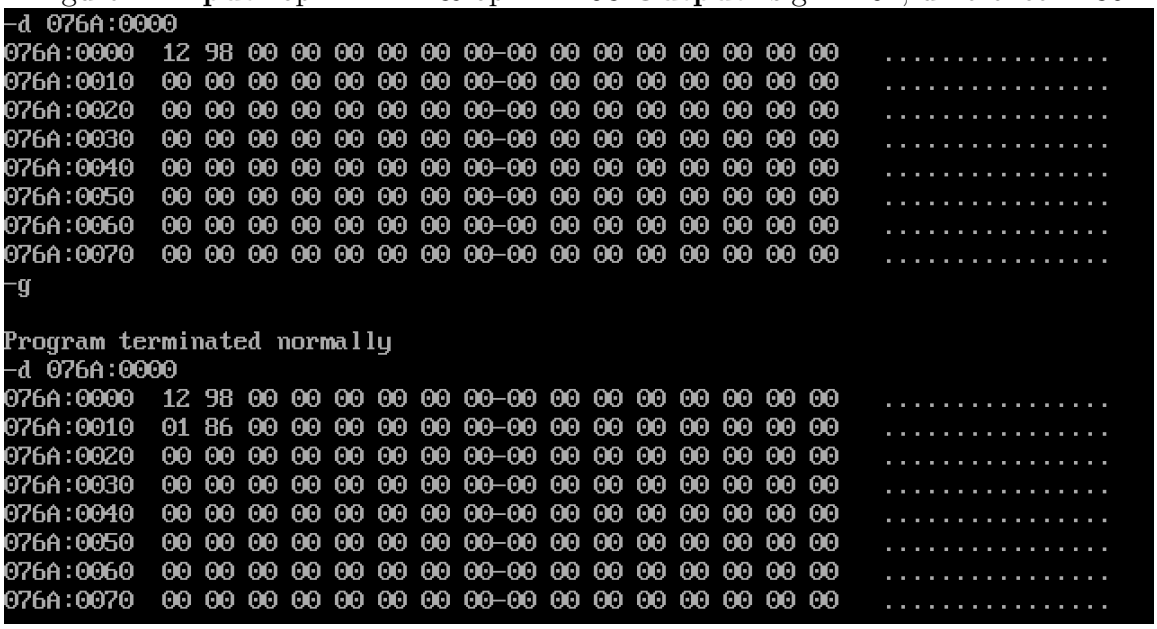
Program		Comments
start:	MOV AX,data	Move data segment address contents to AX register
	MOV ds,AX	Move data in AX register to DS register
	MOV AL, opr1	Move first operand into AL
	MOV BL ,opr2	Move second operand into BL
	SUB AL, BL	Subtract BL from AL
	MOV CL, 00h	Initialise CL to 0
	DAS	Decimal adjust after subtraction
	JNC **stop**	If there is no borrow(carry)
	MOV BL, 99h	Load 99 into BL register
	SUB BL, AL	Subtract AL from 99
	INC BL	BL now contains 10s complement of AL
	DAS	Decimal Adjust after subtraction
	mov AL, BL	Move 10s complement into AL
	MOV CL, 01h	Set sign flag
stop:	MOV difference, AL	Store difference from AL
	MOV sign, CL	Store sign flag from CL
	MOV ah,4ch	
	INT 21h	Request interrupt routine

Unassembled Code:

```
D:\>debug 7B.EXE
-U
076C:0100 B86A07        MOV     AX,076A
076C:0103 8ED8          MOV     DS,AX
076C:0105 A00000        MOV     AL,[0000]
076C:0108 8A1E0100      MOV     BL,[0001]
076C:010C 2AC3          SUB     AL,BL
076C:010E B100          MOV     CL,00
076C:0110 2F            DAS
076C:0111 730B          JNB     011E
076C:0113 B399          MOV     BL,99
076C:0115 2AD8          SUB     BL,AL
076C:0117 FEC3          INC     BL
076C:0119 2F            DAS
076C:011A 8AC3          MOV     AL,BL
076C:011C FEC1          INC     CL
076C:011E A21100        MOV     [0011],AL
```

Input and Output:

Figure 2: **Input:** opr_1 = 12 & opr_2 = 98 **Output:** sign = 01, difference = 86



Result:

8086 ASL programs for BCD addition and subtraction have been executed successfully using MS - DOSBox.