Exercise 7 – BCD Addition and Subtraction
7A – BCD Addition

Aim:

To add two BCD numbers.

Procedure for executing MASM:

1. Mount the local folder in the DOS-BOX using a temp disk name:
   `mount <disk-name> <folder-location>`
2. Change directory into the mounted disk: `<disk-name>: `
3. Assemble the instructions: `masm <file-name>.asm`
4. Link the object file(s) to produce an executable file(.exe): `link <file-name>.obj; ` Note that removal of semi-colon will make linking process interactive.
5. Debug the executable file to read the memory map and execute the program: `debug <file-name>.exe`. After entering debug mode,
   a. `d <segment:offset> ` - dump(read) memory map from the given location
   b. `e <segment:offset> ` - edit memory values from the given location. Use 'White space' to continue editing and 'new line' to exit editing.
   c. `u ` - unassemble code (with or without <segment:offset>)
   d. `g ` - execute the program
   e. `? ` - display command list
   f. `q` - quit the debugger

Algorithm:

1. Initialise data and extra segment using their respective registers.
2. Move the augend into AL.
3. Add the contents of AL with addend and store it in place.
4. On decimal adjust after addition (DAA), we get the BCD adjusted sum at AL and carry at CF.
5. Move the sum at AL to memory, sum.
6. If CF is set, move 01H to carry to indicate generation of carry.
7. Terminate the program.

Program:

Program	Comment

`; 7a - BCD Addtion`
 	
`assume cs: code, ds: data`
 	
`data segment`
`    augend db 99H`
`    addend db 99H`
`    carry db 00H`
`    sum db 00H`
`data ends`
 	
`code segment`
`start:  mov ax, data`
`        mov ds, ax`
 	
`        mov al, augend`
`        add al, addend`
`        daa`
 	
`        mov sum, al`
 	
`        jnc term`
`        mov carry, 01H`
 	
`term:   mov ah, 4cH`
`        int 21H`
`code ends`
`end start`
	Comment after ';'
	
	Map CS to code segment, DS to data segment
	
	Initialise data segment and extra segment
	db = define a byte
	Initialise augend and addend
	Initialise carry and sum with 00H
	
	Initialise code segment
	Move the starting address of data segment in ax, then move ax to ds.
	
	Move augend to AL
	Add AL with addend (AL = AL + Addend)
	Decimal Adjust after Addition
	
	Move AL to sum
	
	If CF not set, terminate, else,
	Move 01H to Carry
	
	Set ah = 4cH
	Call interrupt routine 21H for DOS, which terminates if ah = 4cH

Unassembled code:

```
D:\>debug add.exe
-u
076B:0000 B86A07        MOV     AX,076A
076B:0003 8ED8          MOV     DS,AX
076B:0005 A00000        MOV     AL,[0000]
076B:0008 02060100      ADD     AL,[0001]
076B:000C 27            DAA
076B:000D A20300        MOV     [0003],AL
076B:0010 7305          JNB     0017
076B:0012 C606020001    MOV     BYTE PTR [0002],01
076B:0017 B44C          MOV     AH,4C
076B:0019 CD21          INT     21
```

Snapshot of sample input and output:

Case i: No carry
Augend = 31 Addend = 68
Carry = 00 Sum   = 99

```
-d 076a:0000
076A:0000   31 68 00 00 00 00 00 00-00 00 00 00 00 00 00 00    1h..............
076A:0010   B8 6A 07 8E D8 A0 00 00-02 06 01 00 27 A2 03 00    .j.........'...
076A:0020   73 05 C6 06 02 00 01 B4-4C CD 21 F8 EB 1E 8A 5E    s.......L.!....^
076A:0030   F9 B7 00 D1 E3 8B 87 AE-16 3B 46 FE 77 09 89 46    .........;F.w..F
076A:0040   FE 8A 46 F9 88 46 F8 FE-46 F9 EB C9 8A 5E F8 B7    ..F..F..F....^..
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   31 68 00 99 00 00 00 00-00 00 00 00 00 00 00 00    1h..............
076A:0010   B8 6A 07 8E D8 A0 00 00-02 06 01 00 27 A2 03 00    .j.........'...
076A:0020   73 05 C6 06 02 00 01 B4-4C CD 21 F8 EB 1E 8A 5E    s.......L.!....^
076A:0030   F9 B7 00 D1 E3 8B 87 AE-16 3B 46 FE 77 09 89 46    .........;F.w..F
076A:0040   FE 8A 46 F9 88 46 F8 FE-46 F9 EB C9 8A 5E F8 B7    ..F..F..F....^..
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.
```

Case ii: Carry generated
Augend = 99 Addend = 99
Carry = 01 Sum   = 98

```
-d 076a:0000
076A:0000   99 99 00 00 00 00 00 00-00 00 00 00 00 00 00 00    ..............
076A:0010   B8 6A 07 8E D8 A0 00 00-02 06 01 00 27 A2 03 00    .j.........'...
076A:0020   73 05 C6 06 02 00 01 B4-4C CD 21 F8 EB 1E 8A 5E    s.......L.!....^
076A:0030   F9 B7 00 D1 E3 8B 87 AE-16 3B 46 FE 77 09 89 46    .........;F.w..F
076A:0040   FE 8A 46 F9 88 46 F8 FE-46 F9 EB C9 8A 5E F8 B7    ..F..F..F....^..
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   99 99 01 98 00 00 00 00-00 00 00 00 00 00 00 00    ..............
076A:0010   B8 6A 07 8E D8 A0 00 00-02 06 01 00 27 A2 03 00    .j.........'...
076A:0020   73 05 C6 06 02 00 01 B4-4C CD 21 F8 EB 1E 8A 5E    s.......L.!....^
076A:0030   F9 B7 00 D1 E3 8B 87 AE-16 3B 46 FE 77 09 89 46    .........;F.w..F
076A:0040   FE 8A 46 F9 88 46 F8 FE-46 F9 EB C9 8A 5E F8 B7    ..F..F..F....^..
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.
```

Result:

Program to add 2 BCD numbers is assembled, executed and verified.

7B – BCD Subtraction

Aim:

To subtract two BCD numbers.

Algorithm:

1. Initialise data and extra segment using their respective registers.
2. Move the minuend into AL.
3. Subtract the contents of AL with subtrahend and store it in place.
4. On decimal adjust after addition (DAA), we get the BCD adjusted difference at AL and carry at CF.
5. Move the sum at AL to memory, diff.
6. If CF is not set, jump to term (Step 8).
7. If CF is set,
   a. Move 01H to sign to indicate that minuend was smaller than subtrahend.
   b. Take the 10's complement of the result: move 99H to BL, and subtract AL from BL and increment BL.
   c. Move BL back to AL.
8. Move AL to memory, diff.
9. Terminate the program.

Program:

Program	Comment

`; 7b - BCD subtraction`
 	
`assume cs: code, ds: data`
 	
`data segment`
`    minuend db 38H`
`    subtrahend db 61H`
`    sign db 00H`
`    diff db 00H`
`data ends`
 	
`code segment`
`start:  mov ax, data`
`        mov ds, ax`
 	
`        mov al, minuend`
`        sub al, subtrahend`
 	
`        das`
 	
`        jnc term`
`        mov sign, 01H`
 	
`        mov bl, 99H`
`        sub bl, al`
`        inc bl`
`        mov al, bl`
 	
`term:   mov diff, al`
 	
`        mov ah,4cH`
`        int 21H`
`code ends`
`end start`
	Comment after ';'
	
	Map CS to code segment, DS to data segment
	
	Initialise data segment and extra segment
	db = define a byte
	Initialise minuend and subtrahend
	Initialise sign and diff with 00H
	
	Initialise code segment
	Move the starting address of data segment in ax, then move ax to ds.
	
	Move minuend to AL
	Add AL with subtrahend (AL = AL - Subtrahend)
	
	Decimal Adjust after Subtraction
	
	If CF not set, jump to term, else, Move 01H to Sign
	
	Move 99H to BL
	Subtract AL from BL (BL = BL – AL)
	Increment BL
	Move BL back to AL
	
	Move AL to memory, diff
	
	Set ah = 4cH
	Call interrupt routine 21H for DOS, which terminates if ah = 4cH

Unassembled code:

```
D:\>debug sub.exe
-u
076B:0000 B86A07          MOV     AX,076A
076B:0003 8ED8            MOV     DS,AX
076B:0005 A00000          MOV     AL,[0000]
076B:0008 2A060100        SUB     AL,[0001]
076B:000C 2F              DAS
076B:000D 730D            JNB     001C
076B:000F C606020001      MOV     BYTE PTR [0002],01
076B:0014 B399            MOV     BL,99
076B:0016 2AD8            SUB     BL,AL
076B:0018 FEC3            INC     BL
076B:001A 8AC3            MOV     AL,BL
076B:001C A20300          MOV     [0003],AL
076B:001F B44C            MOV     AH,4C
076B:0021 CD21            INT     21
```

Snapshot of sample input and output:

Case i: Minuend > Subtrahend
Minuend = 00H Subtrahend = 00H
Sign   = 00H Difference = 00H

```
-d 076a:0000
076A:0000   61 38 00 00 00 00 00 00-00 00 00 00 00 00 00 00   a8..............
076A:0010   B8 6A 07 8E D8 A0 00 00-2A 06 01 00 2F 73 0D C6   .j......*.../s..
076A:0020   06 02 00 01 B3 99 2A D8-FE C3 8A C3 A2 03 00 B4   ......*.........
076A:0030   4C CD 21 D1 E3 8B 87 AE-16 3B 46 FE 77 09 89 46   L.!......;F.w..F
076A:0040   FE 8A 46 F9 88 46 F8 FE-46 F9 EB C9 8A 5E F8 B7   ..F..F..F....^..
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   61 38 00 23 00 00 00 00-00 00 00 00 00 00 00 00   a8.#............
076A:0010   B8 6A 07 8E D8 A0 00 00-2A 06 01 00 2F 73 0D C6   .j......*.../s..
076A:0020   06 02 00 01 B3 99 2A D8-FE C3 8A C3 A2 03 00 B4   ......*.........
076A:0030   4C CD 21 D1 E3 8B 87 AE-16 3B 46 FE 77 09 89 46   L.!......;F.w..F
076A:0040   FE 8A 46 F9 88 46 F8 FE-46 F9 EB C9 8A 5E F8 B7   ..F..F..F....^..
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.
```

Department of Computer Science and Engineering

```
Case ii: Minuend < Subtrahend
Minuend = 00H Subtrahend = 00H
Sign    = 00H Difference = 00H
```

```
-d 076a:0000
076A:0000   38 61 00 00 00 00 00 00-00 00 00 00 00 00 00 00   8a..............
076A:0010   B8 6A 07 8E D8 A0 00 00-2A 06 01 00 2F 73 0D C6   .j......*.../s..
076A:0020   06 02 00 01 B3 99 2A D8-FE C3 8A C3 A2 03 00 B4   ......*.........
076A:0030   4C CD 21 D1 E3 8B 87 AE-16 3B 46 FE 77 09 89 46   L.!......;F.w..F
076A:0040   FE 8A 46 F9 88 46 F8 FE-46 F9 EB C9 8A 5E F8 B7   ..F..F..F....^..
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   38 61 01 23 00 00 00 00-00 00 00 00 00 00 00 00   8a.#............
076A:0010   B8 6A 07 8E D8 A0 00 00-2A 06 01 00 2F 73 0D C6   .j......*.../s..
076A:0020   06 02 00 01 B3 99 2A D8-FE C3 8A C3 A2 03 00 B4   ......*.........
076A:0030   4C CD 21 D1 E3 8B 87 AE-16 3B 46 FE 77 09 89 46   L.!......;F.w..F
076A:0040   FE 8A 46 F9 88 46 F8 FE-46 F9 EB C9 8A 5E F8 B7   ..F..F..F....^..
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.
```

Result:

Program to subtract 2 BCD numbers is assembled, executed and verified.