Matrix Operations, Sorting, BCD Arithmetic

Expt No:	5, 6, 7	**Name:**	Shivanirudh S G
Date :	09/10/2020	**Reg No:**	185001146

Ex 5: Matrix Operations

Aim:

To perform matrix operations in 8086.

Matrix Addition

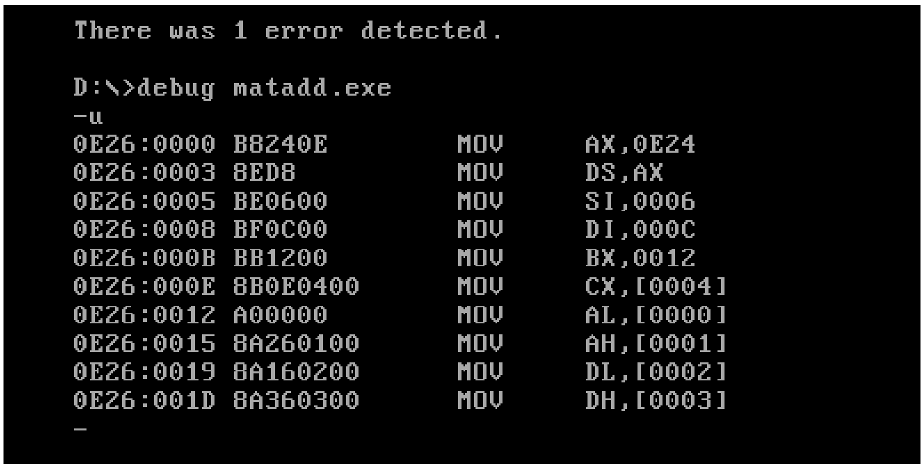
Algorithm:

- Move the data segment to the AX register and then move it to the DS register.
- Move offsets of mat1, mat2 and mat3 into SI, DI, BX registers respectively.
- Move value of count to CX register
- Move values of r1, r2, c1, c2 into AL, AH, DL, DH registers respectively.
- Compare AL, AH by CMP AL, AH and jump to exit if unequal.
- Compare BL, BH by CMP BL, BH and jump to exit if unequal.
- Move value at [SI] to AL register.
- Add AL with value at [DI].
- Move value at AL to [BX].
- Increment SI, DI and BX, decrease CX, repeat till CX = 0.

Program:

Program	**Comments**
assume cs:code, ds:data	Declare code and data segments
data segment	Start of data segment
r1 db 02H	Define byte r1 with value 02H
r2 db 02H	Define byte r2 with value 02H
c1 db 03H	Define byte c1 with value 03H
c2 db 03H	Define byte c2 with value 03H
count dw 0006H	Define word count with value 0006H
mat1 db 22H, 33H, 44H, 55H, 66H, 77H	Define matrix of values mat1
mat2 db 33H, 44H, 55H, 66H, 77H, 88H	Define matrix of values mat2
mat3 db ?	Define result matrix of values mat3
data ends	End of data segment
code segment	Start of code segment
start: mov ax, data	Move data to AX register
mov ds, ax	Move contents of AX register to DS register
mov dl, 0AH	Move hex value 0A to DL register
mov si, offset mat1	Move offset of mat1 to SI register
mov di, offset mat2	Move offset of mat2 to DI register
mov bx, offset mat3	Move offset of mat3 to BX register
mov cx, count	Move value of count to CX register
mov al, r1	Move value of r1 to AL register
mov ah, r2	Move value of r2 to AH register
mov dl, c1	Move value of c1 to DL register
mov dh, c2	Move value of c2 to DH register
cmp al, ah	Compare values of AL and AH registers
jne exit	Jump to exit if ZF = 0
cmp dl, dh	Compare values of DL, DH registers
jne exit	Jump to exit if ZF = 0
here: mov al, [si]	Move contents at SI to AL register
add al, [di]	AL = AL + [DI]
mov [bx], al	Move contents of AL register to BX register
inc si	Increment value in SI register
inc di	Increment value in DI register
inc bx	Increment value in BX register
dec cx	Decrement value of CX register
jnz here	Jump to here if ZF = 0
exit: mov ah, 4ch	To request interrupt
int 21h	Request interrupt routine
code ends	End of code segment
end start	

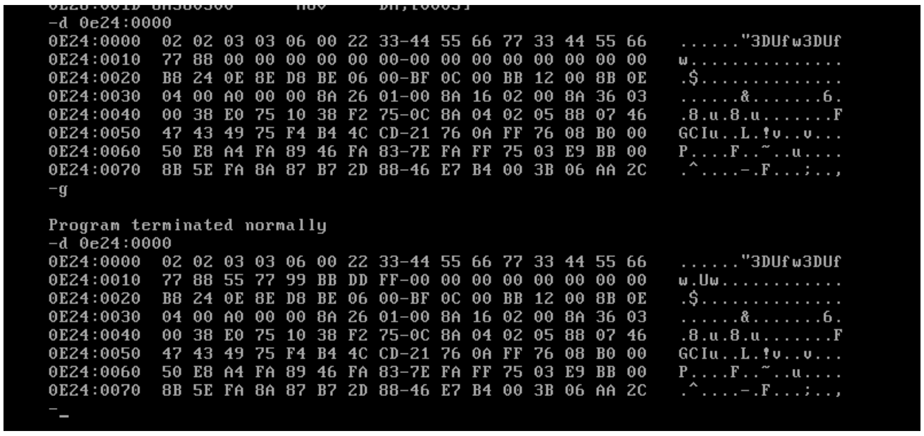
Unassembled code:



```
There was 1 error detected.

D:\>debug matadd.exe
-u
0E26:0000 B8240E        MOV     AX,0E24
0E26:0003 8ED8          MOV     DS,AX
0E26:0005 BE0600        MOV     SI,0006
0E26:0008 BF0C00        MOV     DI,000C
0E26:000B BB1200        MOV     BX,0012
0E26:000E 8B0E0400      MOV     CX,[0004]
0E26:0012 A00000        MOV     AL,[0000]
0E26:0015 8A260100      MOV     AH,[0001]
0E26:0019 8A160200      MOV     DL,[0002]
0E26:001D 8A360300      MOV     DH,[0003]
-
```

Input and Output:



```
-d 0e24:0000
0E24:0000  02 02 03 03 06 00 22 33-44 55 66 77 33 44 55 66   ......"3DUfw3DUf
0E24:0010  77 88 00 00 00 00 00 00-00 00 00 00 00 00 00 00   w...............
0E24:0020  B8 24 0E 8E D8 BE 06 00-BF 0C 00 BB 12 00 8B 0E   .$..............
0E24:0030  04 00 A0 00 00 8A 26 01-00 8A 16 02 00 8A 36 03   ......&.......6.
0E24:0040  00 38 E0 75 10 38 F2 75-0C 8A 04 02 05 88 07 46   .8.u.8.u.......F
0E24:0050  47 43 49 75 F4 B4 4C CD-21 76 0A FF 76 08 B0 00   GCIu..L.!v..v...
0E24:0060  50 E8 A4 FA 89 46 FA 83-7E FA FF 75 03 E9 BB 00   P....F..~..u....
0E24:0070  8B 5E FA 8A 87 B7 2D 88-46 E7 B4 00 3B 06 AA 2C   .^....-.F...;..,
-g

Program terminated normally
-d 0e24:0000
0E24:0000  02 02 03 03 06 00 22 33-44 55 66 77 33 44 55 66   ......"3DUfw3DUf
0E24:0010  77 88 55 77 99 BB DD FF-00 00 00 00 00 00 00 00   w.Uw............
0E24:0020  B8 24 0E 8E D8 BE 06 00-BF 0C 00 BB 12 00 8B 0E   .$..............
0E24:0030  04 00 A0 00 00 8A 26 01-00 8A 16 02 00 8A 36 03   ......&.......6.
0E24:0040  00 38 E0 75 10 38 F2 75-0C 8A 04 02 05 88 07 46   .8.u.8.u.......F
0E24:0050  47 43 49 75 F4 B4 4C CD-21 76 0A FF 76 08 B0 00   GCIu..L.!v..v...
0E24:0060  50 E8 A4 FA 89 46 FA 83-7E FA FF 75 03 E9 BB 00   P....F..~..u....
0E24:0070  8B 5E FA 8A 87 B7 2D 88-46 E7 B4 00 3B 06 AA 2C   .^....-.F...;..,
-_
```

Figure 1: **Input:** *mat1:* 22H, 33H, 44H, 55H, 66H, 77H; *mat2:* 33H, 44H, 55H, 66H, 77H, 88H ;
Output: *mat3:* 55H, 77H, 99H, BBH, DDH, FFH

Matrix Subtraction

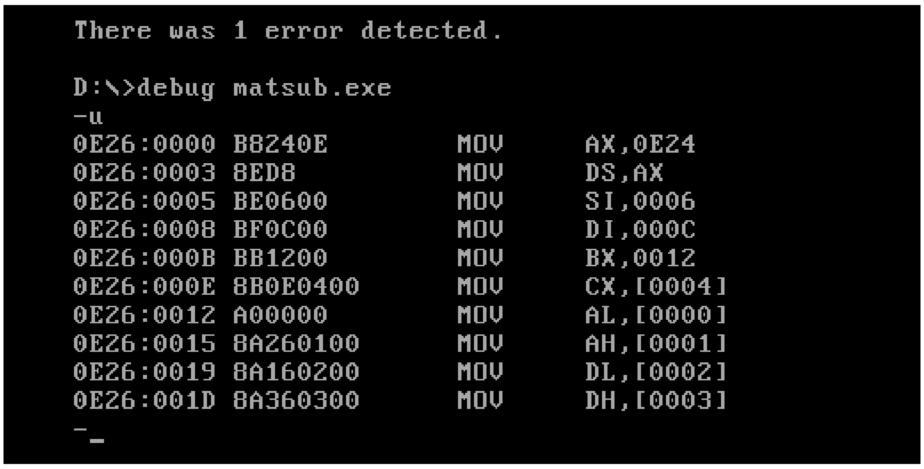
Algorithm:

- Move the data segment to the AX register and then move it to the DS register.
- Move offsets of mat1, mat2 and mat3 into SI, DI, BX registers respectively.
- Move value of count to CX register
- Move values of r1, r2, c1, c2 into AL, AH, DL, DH registers respectively.
- Compare AL, AH by CMP AL, AH and jump to exit if unequal.
- Compare BL, BH by CMP BL, BH and jump to exit if unequal.
- Move value at [DI] to AL register.
- Subtract AL with value at [SI].
- Move value at AL to [BX].
- Increment SI, DI and BX, decrease CX, repeat till CX = 0.

Program:

Program	Comments
assume cs:code, ds:data	Declare code and data segments
data segment	Start of data segment
r1 db 02H	Define byte r1 with value 02H
r2 db 02H	Define byte r2 with value 02H
c1 db 03H	Define byte c1 with value 03H
c2 db 03H	Define byte c2 with value 03H
count dw 0006H	Define word count with value 0006H
mat1 db 22H, 33H, 44H, 55H, 66H, 77H	Define matrix of values mat1
mat2 db 33H, 44H, 55H, 66H, 77H, 88H	Define matrix of values mat2
mat3 db ?	Define result matrix of values mat3
data ends	End of data segment
code segment	Start of code segment
start: mov ax, data	Move data to AX register
mov ds, ax	Move contents of AX register to DS register
mov dl, 0AH	Move hex value 0A to DL register
mov si, offset mat1	Move offset of mat1 to SI register
mov di, offset mat2	Move offset of mat2 to DI register
mov bx, offset mat3	Move offset of mat3 to BX register
mov cx, count	Move value of count to CX register
mov al, r1	Move value of r1 to AL register
mov ah, r2	Move value of r2 to AH register
mov dl, c1	Move value of c1 to DL register
mov dh, c2	Move value of c2 to DH register
cmp al, ah	Compare values of AL and AH registers
jne exit	Jump to exit if ZF = 0
cmp dl, dh	Compare values of DL, DH registers
jne exit	Jump to exit if ZF = 0
here: mov al, [di]	Move contents at DI to AL register
add al, [si]	AL = AL - [SI]
mov [bx], al	Move contents of AL register to BX register
inc si	Increment value in SI register
inc di	Increment value in DI register
inc bx	Increment value in BX register
dec cx	Decrement value of CX register
jnz here	Jump to here if ZF = 0
exit: mov ah, 4ch	To request interrupt
int 21h	Request interrupt routine
code ends	End of code segment
end start	

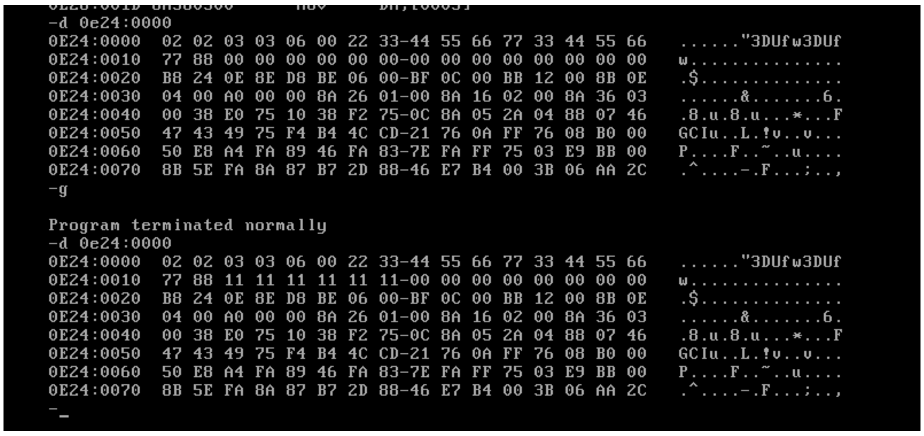
Unassembled code:



```
There was 1 error detected.

D:\>debug matsub.exe
-u
0E26:0000 B8240E        MOV     AX,0E24
0E26:0003 8ED8          MOV     DS,AX
0E26:0005 BE0600        MOV     SI,0006
0E26:0008 BF0C00        MOV     DI,000C
0E26:000B BB1200        MOV     BX,0012
0E26:000E 8B0E0400      MOV     CX,[0004]
0E26:0012 A00000        MOV     AL,[0000]
0E26:0015 8A260100      MOV     AH,[0001]
0E26:0019 8A160200      MOV     DL,[0002]
0E26:001D 8A360300      MOV     DH,[0003]
-_
```

Input and Output:



```
-d 0e24:0000
0E24:0000  02 02 03 03 06 00 22 33-44 55 66 77 33 44 55 66   ......"3DUfw3DUf
0E24:0010  77 88 00 00 00 00 00 00-00 00 00 00 00 00 00 00   w...............
0E24:0020  B8 24 0E 8E D8 BE 06 00-BF 0C 00 BB 12 00 8B 0E   .$..............
0E24:0030  04 00 A0 00 00 8A 26 01-00 8A 16 02 00 8A 36 03   ......&.......6.
0E24:0040  00 38 E0 75 10 38 F2 75-0C 8A 05 2A 04 88 07 46   .8.u.8.u...*...F
0E24:0050  47 43 49 75 F4 B4 4C CD-21 76 0A FF 76 08 B0 00   GCIu..L.!v..v...
0E24:0060  50 E8 A4 FA 89 46 FA 83-7E FA FF 75 03 E9 BB 00   P....F..~..u....
0E24:0070  8B 5E FA 8A 87 B7 2D 88-46 E7 B4 00 3B 06 AA 2C   .^....-.F...;..,
-g

Program terminated normally
-d 0e24:0000
0E24:0000  02 02 03 03 06 00 22 33-44 55 66 77 33 44 55 66   ......"3DUfw3DUf
0E24:0010  77 88 11 11 11 11 11 11-00 00 00 00 00 00 00 00   w...............
0E24:0020  B8 24 0E 8E D8 BE 06 00-BF 0C 00 BB 12 00 8B 0E   .$..............
0E24:0030  04 00 A0 00 00 8A 26 01-00 8A 16 02 00 8A 36 03   ......&.......6.
0E24:0040  00 38 E0 75 10 38 F2 75-0C 8A 05 2A 04 88 07 46   .8.u.8.u...*...F
0E24:0050  47 43 49 75 F4 B4 4C CD-21 76 0A FF 76 08 B0 00   GCIu..L.!v..v...
0E24:0060  50 E8 A4 FA 89 46 FA 83-7E FA FF 75 03 E9 BB 00   P....F..~..u....
0E24:0070  8B 5E FA 8A 87 B7 2D 88-46 E7 B4 00 3B 06 AA 2C   .^....-.F...;..,
-_
```

Figure 2: **Input:** *mat1:* 22H, 33H, 44H, 55H, 66H, 77H; *mat2:* 33H, 44H, 55H, 66H, 77H, 88H ;
Output: *mat3:* 11H, 11H, 11H, 11H, 11H, 11H

Result:

The 8086 programs were written to perform matrix operations, and the results observed.

Ex 6: Sorting Operations

Aim:

To perform sorting operations in 8086.

Ascending Order

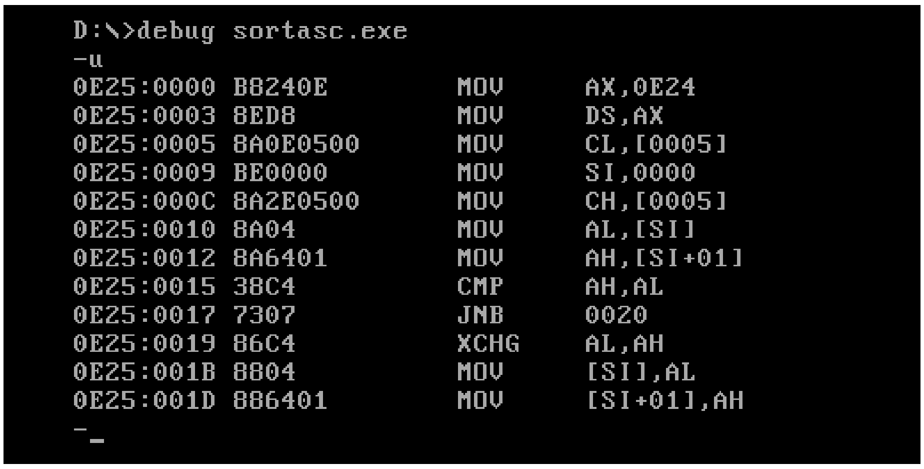
Algorithm:

- Move the data segment to the AX register and then move it to the DS register.
- Move value of count to CL register.
- Move offset of arr into SI register under label OUTER.
- Move value of count to CH register.
- Move value at [SI] to AL register, [SI+1] to AH register, under label INNER.
- Compare AH, AL with CMP AH, AL.
- If CF = 0, jump to label NOSWAP.
- Swap values of AH, AL with XCHG AH, AL
- Move value in AL to [SI] register, AH to [SI+1].
- Increment SI, decrement CH under label NOSWAP.
- Jump to INNER if ZF = 0.
- Decrement CL and jump to OUTER if ZF = 0.

Program:

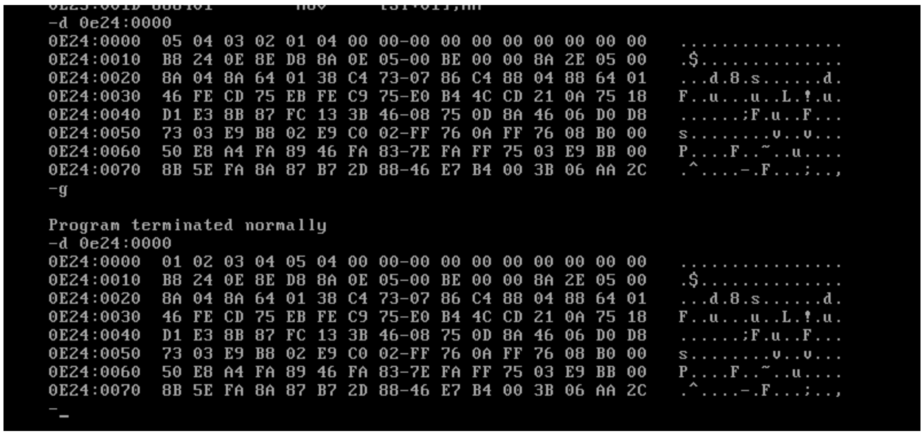
Program	**Comments**
assume cs:code, ds:data	Declare code and data segments
data segment	Start of data segment
arr db 05H, 04H, 03H, 02H, 01H	Define array of values arr
count db 04H	Define byte count with hex value 04
data ends	End of data segment
code segment	Start of code segment
start: mov ax, data	Move data to AX register
mov ds, ax	Move contents of AX register to DS register
mov cl, count	Move value of count to CL register
outer: mov si, offset arr	Move offset of arr to SI register
mov ch, count	Move value of count to CH register
inner: mov al, [si]	Move value at offset in SI to AL register
mov ah, [si+1]	Move value at offset in SI register +1 to AH
cmp ah, al	Compare values in AH, AL registers
jnc noswap	Jump to NOSWAP if CF = 0
xchg al, ah	Swap values in AL, AH registers
mov [si], al	Move value in AL register to offset at [SI]
mov [si+1], ah	Move value in AH register to offset at [SI]+1
noswap: inc si	Increment value of SI
dec ch	Decrement value of CH
jnz inner	Jump to INNER if ZF = 0
dec cl	Decrement value of CL
jnz outer	Jump to OUTER if ZF = 0
mov ah, 4ch	To request interrupt
int 21h	Request interrupt routine
code ends	End of code segment
end start	

Unassembled code:



```
D:\>debug sortasc.exe
-u
0E25:0000 B8240E        MOV     AX,0E24
0E25:0003 8ED8          MOV     DS,AX
0E25:0005 8A0E0500      MOV     CL,[0005]
0E25:0009 BE0000        MOV     SI,0000
0E25:000C 8A2E0500      MOV     CH,[0005]
0E25:0010 8A04          MOV     AL,[SI]
0E25:0012 8A6401        MOV     AH,[SI+01]
0E25:0015 38C4          CMP     AH,AL
0E25:0017 7307          JNB     0020
0E25:0019 86C4          XCHG    AL,AH
0E25:001B 8804          MOV     [SI],AL
0E25:001D 886401        MOV     [SI+01],AH
-_
```

Input and Output:



```
-d 0e24:0000
0E24:0000  05 04 03 02 01 04 00 00-00 00 00 00 00 00 00 00   ................
0E24:0010  B8 24 0E 8E D8 8A 0E 05-00 BE 00 00 8A 2E 05 00   .$..............
0E24:0020  8A 04 8A 64 01 38 C4 73-07 86 C4 88 04 88 64 01   ...d.8.s......d.
0E24:0030  46 FE CD 75 EB FE C9 75-E0 B4 4C CD 21 0A 75 18   F..u...u..L.!.u.
0E24:0040  D1 E3 8B 87 FC 13 3B 46-08 75 0D 8A 46 06 D0 D8   ......;F.u..F...
0E24:0050  73 03 E9 B8 02 E9 C0 02-FF 76 0A FF 76 08 B0 00   s........v..v...
0E24:0060  50 E8 A4 FA 89 46 FA 83-7E FA FF 75 03 E9 BB 00   P....F..~..u....
0E24:0070  8B 5E FA 8A 87 B7 2D 88-46 E7 B4 00 3B 06 AA 2C   .^....-.F...;..,
-g

Program terminated normally
-d 0e24:0000
0E24:0000  01 02 03 04 05 04 00 00-00 00 00 00 00 00 00 00   ................
0E24:0010  B8 24 0E 8E D8 8A 0E 05-00 BE 00 00 8A 2E 05 00   .$..............
0E24:0020  8A 04 8A 64 01 38 C4 73-07 86 C4 88 04 88 64 01   ...d.8.s......d.
0E24:0030  46 FE CD 75 EB FE C9 75-E0 B4 4C CD 21 0A 75 18   F..u...u..L.!.u.
0E24:0040  D1 E3 8B 87 FC 13 3B 46-08 75 0D 8A 46 06 D0 D8   ......;F.u..F...
0E24:0050  73 03 E9 B8 02 E9 C0 02-FF 76 0A FF 76 08 B0 00   s........v..v...
0E24:0060  50 E8 A4 FA 89 46 FA 83-7E FA FF 75 03 E9 BB 00   P....F..~..u....
0E24:0070  8B 5E FA 8A 87 B7 2D 88-46 E7 B4 00 3B 06 AA 2C   .^....-.F...;..,
-_
```

Figure 3: **Input:** 05H, 04H, 03H, 02H, 01H;
Output: 01H, 02H, 03H, 04H, 05H

Descending Order

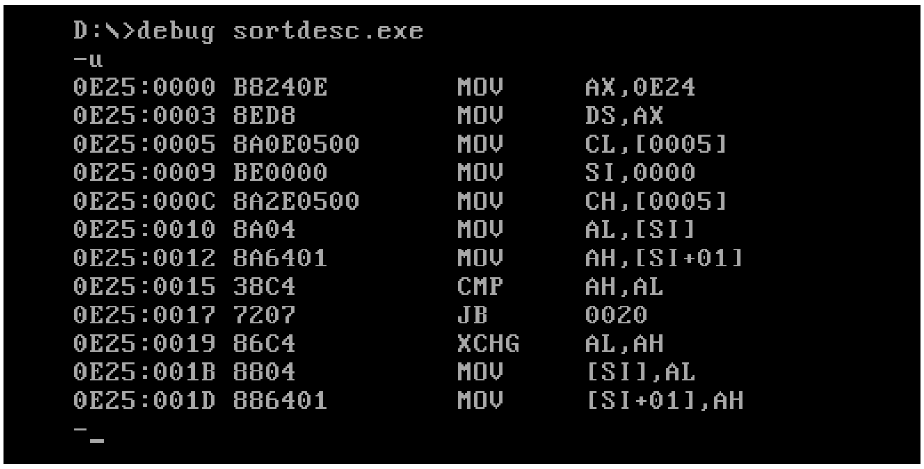
Algorithm:

- Move the data segment to the AX register and then move it to the DS register.
- Move value of count to CL register.
- Move offset of arr into SI register under label OUTER.
- Move value of count to CH register.
- Move value at [SI] to AL register, [SI+1] to AH register, under label INNER.
- Compare AH, AL with CMP AH, AL.
- If CF = 1, jump to label NOSWAP.
- Swap values of AH, AL with XCHG AH, AL
- Move value in AL to [SI] register, AH to [SI+1].
- Increment SI, decrement CH under label NOSWAP.
- Jump to INNER if ZF = 0.
- Decrement CL and jump to OUTER if ZF = 0.

Program:

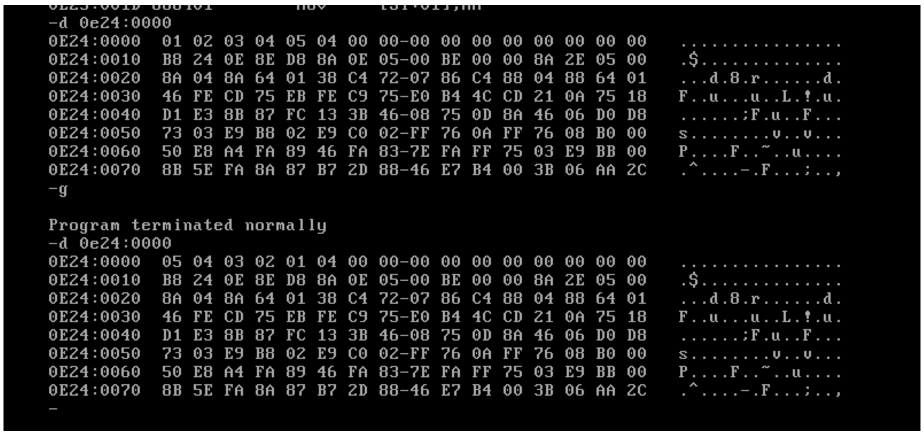
Program	**Comments**
assume cs:code, ds:data	Declare code and data segments
data segment	Start of data segment
arr db 05H, 04H, 03H, 02H, 01H	Define array of values arr
count db 04H	Define byte count with hex value 04
data ends	End of data segment
code segment	Start of code segment
start: mov ax, data	Move data to AX register
mov ds, ax	Move contents of AX register to DS register
mov cl, count	Move value of count to CL register
outer: mov si, offset arr	Move offset of arr to SI register
mov ch, count	Move value of count to CH register
inner: mov al, [si]	Move value at offset in SI to AL register
mov ah, [si+1]	Move value at offset in SI register +1 to AH
cmp ah, al	Compare values in AH, AL registers
jc noswap	Jump to NOSWAP if CF = 1
xchg al, ah	Swap values in AL, AH registers
mov [si], al	Move value in AL register to offset at [SI]
mov [si+1], ah	Move value in AH register to offset at [SI]+1
noswap: inc si	Increment value of SI
dec ch	Decrement value of CH
jnz inner	Jump to INNER if ZF = 0
dec cl	Decrement value of CL
jnz outer	Jump to OUTER if ZF = 0
mov ah, 4ch	To request interrupt
int 21h	Request interrupt routine
code ends	End of code segment
end start	

Unassembled code:



```
D:\>debug sortdesc.exe
-u
0E25:0000 B8240E        MOV     AX,0E24
0E25:0003 8ED8          MOV     DS,AX
0E25:0005 8A0E0500      MOV     CL,[0005]
0E25:0009 BE0000        MOV     SI,0000
0E25:000C 8A2E0500      MOV     CH,[0005]
0E25:0010 8A04          MOV     AL,[SI]
0E25:0012 8A6401        MOV     AH,[SI+01]
0E25:0015 38C4          CMP     AH,AL
0E25:0017 7207          JB      0020
0E25:0019 86C4          XCHG    AL,AH
0E25:001B 8804          MOV     [SI],AL
0E25:001D 886401        MOV     [SI+01],AH
-
```

Input and Output:



```
-d 0e24:0000
0E24:0000  01 02 03 04 05 04 00 00-00 00 00 00 00 00 00 00   ................
0E24:0010  B8 24 0E 8E D8 8A 0E 05-00 BE 00 00 8A 2E 05 00   .$..............
0E24:0020  8A 04 8A 64 01 38 C4 72-07 86 C4 88 04 88 64 01   ...d.8.r......d.
0E24:0030  46 FE CD 75 EB FE C9 75-E0 B4 4C CD 21 0A 75 18   F..u...u..L.!.u.
0E24:0040  D1 E3 8B 87 FC 13 3B 46-08 75 0D 8A 46 06 D0 D8   ......;F.u..F...
0E24:0050  73 03 E9 B8 02 E9 C0 02-FF 76 0A FF 76 08 B0 00   s........v..v...
0E24:0060  50 E8 A4 FA 89 46 FA 83-7E FA FF 75 03 E9 BB 00   P....F..~..u....
0E24:0070  8B 5E FA 8A 87 B7 2D 88-46 E7 B4 00 3B 06 AA 2C   .^....-.F...;..,
-g

Program terminated normally
-d 0e24:0000
0E24:0000  05 04 03 02 01 04 00 00-00 00 00 00 00 00 00 00   ................
0E24:0010  B8 24 0E 8E D8 8A 0E 05-00 BE 00 00 8A 2E 05 00   .$..............
0E24:0020  8A 04 8A 64 01 38 C4 72-07 86 C4 88 04 88 64 01   ...d.8.r......d.
0E24:0030  46 FE CD 75 EB FE C9 75-E0 B4 4C CD 21 0A 75 18   F..u...u..L.!.u.
0E24:0040  D1 E3 8B 87 FC 13 3B 46-08 75 0D 8A 46 06 D0 D8   ......;F.u..F...
0E24:0050  73 03 E9 B8 02 E9 C0 02-FF 76 0A FF 76 08 B0 00   s........v..v...
0E24:0060  50 E8 A4 FA 89 46 FA 83-7E FA FF 75 03 E9 BB 00   P....F..~..u....
0E24:0070  8B 5E FA 8A 87 B7 2D 88-46 E7 B4 00 3B 06 AA 2C   .^....-.F...;..,
-
```

Figure 4: **Input:** 01H, 02H, 03H, 04H, 05H;
Output: 05H, 04H, 03H, 02H, 01H

Result:

The 8086 programs were written to perform matrix operations, and the results observed.

Ex 7: BCD Addition and Subtraction

Aim:

To perform BCD addition and subtraction operations in 8086.

BCD Addition

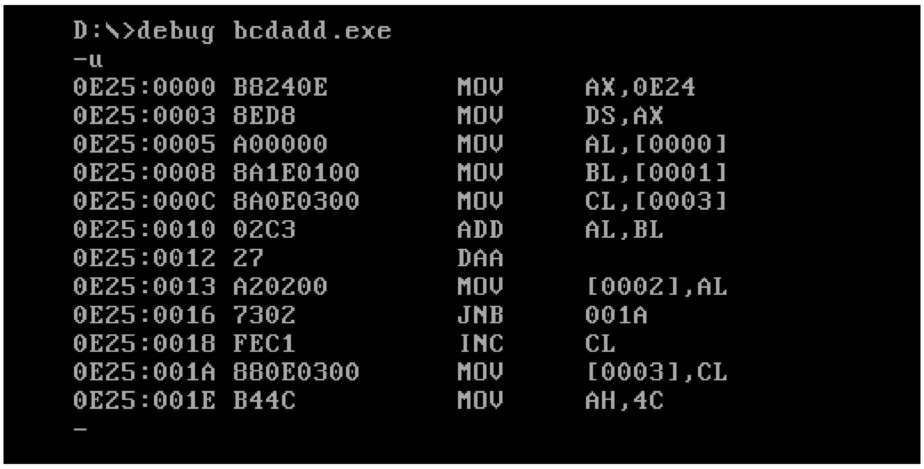
Algorithm:

- Move the data segment to the AX register and then move it to the DS register.
- Move value of num1 to AL, num2 to BL, carry to CL registers.
- Add AL and BL using ADD AL, BL.
- Perform Decimal Adjust After Addition using DAA instruction.
- Move value of AL to ans.
- Jump to label HERE if no carry.
- Increment value of CL.
- Move value of CL to carry, under label HERE.

Program:

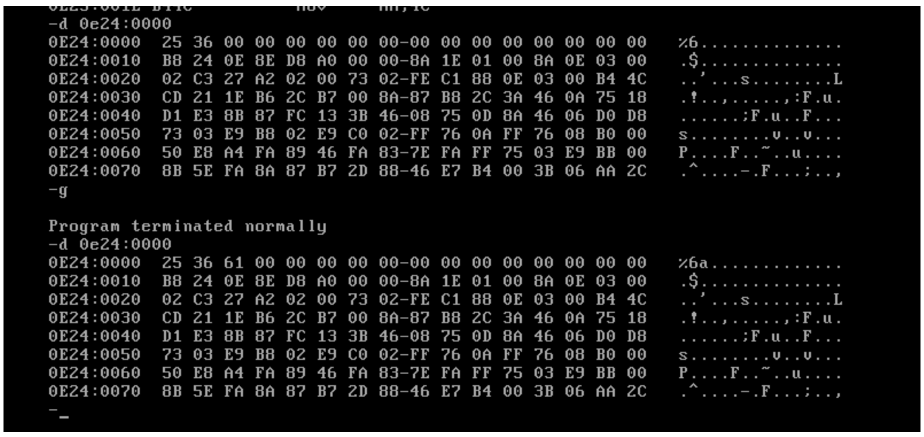
Program	**Comments**
assume cs:code, ds:data	Declare code and data segments
data segment	Start of data segment
num1 db 25H	Define byte num1 with value 25
num2 db 36H	Define byte num2 with value 36
ans db ?	Define byte ans for result
carry db 00H	Define byte carry with value 00
data ends	End of data segment
code segment	Start of code segment
start: mov ax, data	Move data to AX register
mov ds, ax	Move contents of AX register to DS register
mov al, num1	Move value of num1 to AL register
mov bl, num2	Move value of num2 to BL register
mov cl, carry	Move value of carry to CL register
add al, bl	AL = AL + BL
daa	Decimal Adjust after Addition
mov ans, al	Move value of AL register into ans
jnc here	Jump to label HERE if no carry
inc cl	Increment value of CL
here: mov carry, cl	Move value of CL register into carry
mov ah, 4ch	To request interrupt
int 21h	Request interrupt routine
code ends	End of code segment
end start	

Unassembled code:



```
D:\>debug bcdadd.exe
-u
0E25:0000 B8240E        MOV     AX,0E24
0E25:0003 8ED8          MOV     DS,AX
0E25:0005 A00000        MOV     AL,[0000]
0E25:0008 8A1E0100      MOV     BL,[0001]
0E25:000C 8A0E0300      MOV     CL,[0003]
0E25:0010 02C3          ADD     AL,BL
0E25:0012 27            DAA
0E25:0013 A20200        MOV     [0002],AL
0E25:0016 7302          JNB     001A
0E25:0018 FEC1          INC     CL
0E25:001A 880E0300      MOV     [0003],CL
0E25:001E B44C          MOV     AH,4C
-
```

Input and Output:



```
-d 0e24:0000
0E24:0000  25 36 00 00 00 00 00 00-00 00 00 00 00 00 00 00   %6..............
0E24:0010  B8 24 0E 8E D8 A0 00 00-8A 1E 01 00 8A 0E 03 00   .$..............
0E24:0020  02 C3 27 A2 02 00 73 02-FE C1 88 0E 03 00 B4 4C   ..'...s........L
0E24:0030  CD 21 1E B6 2C B7 00 8A-87 B8 2C 3A 46 0A 75 18   .!..,.....,:F.u.
0E24:0040  D1 E3 8B 87 FC 13 3B 46-08 75 0D 8A 46 06 D0 D8   ......;F.u..F...
0E24:0050  73 03 E9 B8 02 E9 C0 02-FF 76 0A FF 76 08 B0 00   s........v..v...
0E24:0060  50 E8 A4 FA 89 46 FA 83-7E FA FF 75 03 E9 BB 00   P....F..~..u....
0E24:0070  8B 5E FA 8A 87 B7 2D 88-46 E7 B4 00 3B 06 AA 2C   .^....-.F...;..,
-g

Program terminated normally
-d 0e24:0000
0E24:0000  25 36 61 00 00 00 00 00-00 00 00 00 00 00 00 00   %6a.............
0E24:0010  B8 24 0E 8E D8 A0 00 00-8A 1E 01 00 8A 0E 03 00   .$..............
0E24:0020  02 C3 27 A2 02 00 73 02-FE C1 88 0E 03 00 B4 4C   ..'...s........L
0E24:0030  CD 21 1E B6 2C B7 00 8A-87 B8 2C 3A 46 0A 75 18   .!..,.....,:F.u.
0E24:0040  D1 E3 8B 87 FC 13 3B 46-08 75 0D 8A 46 06 D0 D8   ......;F.u..F...
0E24:0050  73 03 E9 B8 02 E9 C0 02-FF 76 0A FF 76 08 B0 00   s........v..v...
0E24:0060  50 E8 A4 FA 89 46 FA 83-7E FA FF 75 03 E9 BB 00   P....F..~..u....
0E24:0070  8B 5E FA 8A 87 B7 2D 88-46 E7 B4 00 3B 06 AA 2C   .^....-.F...;..,
-_
```

Figure 5: **Input:** num1: 25, num2: 36; **Output:** ans: 61, carry: 0

BCD Subtraction

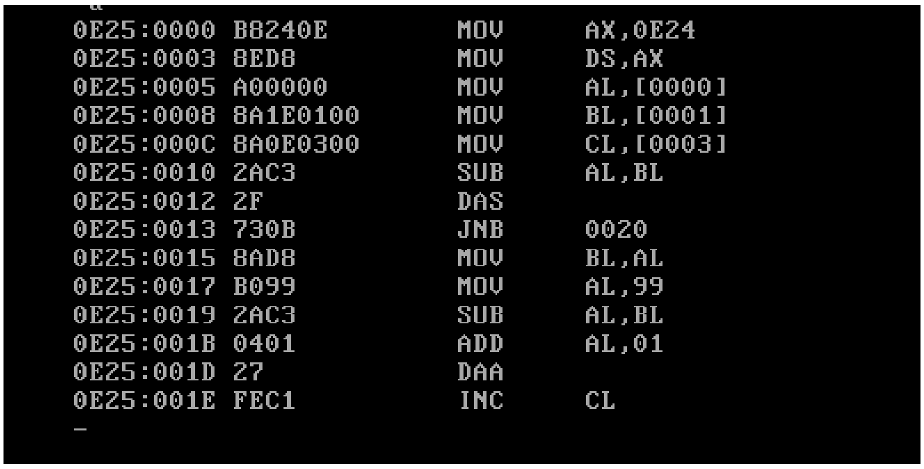
Algorithm:

- Move the data segment to the AX register and then move it to the DS register.
- Move value of num1 to AL, num2 to BL, sign to CL registers.
- Subtract AL and BL using SUB AL, BL.
- Perform Decimal Adjust After Subtraction using DAS instruction.
- Jump to label HERE if no carry.
- Move value in AL to BL register, 99H to AL register.
- Subtract AL and BL using SUB AL, BL.
- Add 1 to AL using ADD AL, 01H.
- Perform Decimal Adjust after Addition using DAA instruction.
- Increment value of CL.
- Move value of CL to sign, under label HERE.
- Move value of AL to ans.

Program:

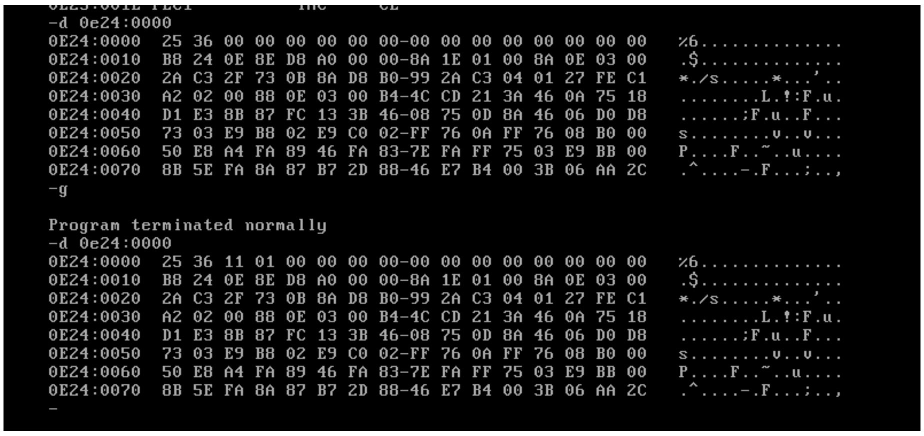
Program	Comments
assume cs:code, ds:data	Declare code and data segments
data segment	Start of data segment
num1 db 25H	Define byte num1 with value 25
num2 db 36H	Define byte num2 with value 36
ans db ?	Define byte ans for result
sign db 00H	Define byte sign with value 00
data ends	End of data segment
code segment	Start of code segment
start: mov ax, data	Move data to AX register
mov ds, ax	Move contents of AX register to DS register
mov al, num1	Move value of num1 to AL register
mov bl, num2	Move value of num2 to BL register
mov cl, sign	Move value of sign to CL register
sub al, bl	AL = AL - BL
das	Decimal Adjust after Subtraction
jnc here	Jump to label HERE if CF = 0
mov bl, al	Move value of AL to BL register
mov al, 99H	Move hex value 99H to AL register
sub al, bl	AL = AL - BL
add al, 01H	AL = AL + 1
daa	Decimal Adjust after Addition
inc cl	Increment value of CL
here: mov ans, al	Move value of AL to ans
mov sign, cl	Move value of CL to sign
mov ah, 4ch	To request interrupt
int 21h	Request interrupt routine
code ends	End of code segment
end start	

Unassembled code:



```
0E25:0000 B8240E        MOV     AX,0E24
0E25:0003 8ED8          MOV     DS,AX
0E25:0005 A00000        MOV     AL,[0000]
0E25:0008 8A1E0100      MOV     BL,[0001]
0E25:000C 8A0E0300      MOV     CL,[0003]
0E25:0010 2AC3          SUB     AL,BL
0E25:0012 2F            DAS
0E25:0013 730B          JNB     0020
0E25:0015 8AD8          MOV     BL,AL
0E25:0017 B099          MOV     AL,99
0E25:0019 2AC3          SUB     AL,BL
0E25:001B 0401          ADD     AL,01
0E25:001D 27            DAA
0E25:001E FEC1          INC     CL
-
```

Input and Output:



```
-d 0e24:0000
0E24:0000  25 36 00 00 00 00 00 00-00 00 00 00 00 00 00 00   %6..............
0E24:0010  B8 24 0E 8E D8 A0 00 00-8A 1E 01 00 8A 0E 03 00   .$..............
0E24:0020  2A C3 2F 73 0B 8A D8 B0-99 2A C3 04 01 27 FE C1   *./s.....*...'..
0E24:0030  A2 02 00 88 0E 03 00 B4-4C CD 21 3A 46 0A 75 18   ........L.!:F.u.
0E24:0040  D1 E3 8B 87 FC 13 3B 46-08 75 0D 8A 46 06 D0 D8   ......;F.u..F...
0E24:0050  73 03 E9 B8 02 E9 C0 02-FF 76 0A FF 76 08 B0 00   s........v..v...
0E24:0060  50 E8 A4 FA 89 46 FA 83-7E FA FF 75 03 E9 BB 00   P....F..~..u....
0E24:0070  8B 5E FA 8A 87 B7 2D 88-46 E7 B4 00 3B 06 AA 2C   .^....-.F...;..,
-g

Program terminated normally
-d 0e24:0000
0E24:0000  25 36 11 01 00 00 00 00-00 00 00 00 00 00 00 00   %6..............
0E24:0010  B8 24 0E 8E D8 A0 00 00-8A 1E 01 00 8A 0E 03 00   .$..............
0E24:0020  2A C3 2F 73 0B 8A D8 B0-99 2A C3 04 01 27 FE C1   *./s.....*...'..
0E24:0030  A2 02 00 88 0E 03 00 B4-4C CD 21 3A 46 0A 75 18   ........L.!:F.u.
0E24:0040  D1 E3 8B 87 FC 13 3B 46-08 75 0D 8A 46 06 D0 D8   ......;F.u..F...
0E24:0050  73 03 E9 B8 02 E9 C0 02-FF 76 0A FF 76 08 B0 00   s........v..v...
0E24:0060  50 E8 A4 FA 89 46 FA 83-7E FA FF 75 03 E9 BB 00   P....F..~..u....
0E24:0070  8B 5E FA 8A 87 B7 2D 88-46 E7 B4 00 3B 06 AA 2C   .^....-.F...;..,
-
```

Figure 6: **Input:** num1: 25, num2: 36; **Output:** ans: 11, sign: 1

Result:

The 8086 programs were written to perform BCD addition and subtraction operations, and the results observed.