Matrix Operations

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

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 3B 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 3B 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.