

Experiment No 5: Matrix Operations

Date: 04-10-2020

NAME: Gayathri M

REG.NO: 185001050

A. AIM:

Program for Matrix addition.

ALGORITHM:

- Initialize the data segment.
- Move data segment address to ds
- Load row1 to cl, row2 to dl.
- Compare cl and dl and terminate if not equal.
- Load col1 to cl, col2 to dl.
- Compare cl and dl and terminate if not equal.
- Move row2 to al.
- Multiply al with cl and move ax to cx.
- Move offset of matrix1 to si, matrix2 to di, result to bx
- Loop here:
 - Move contents pointed by si to al and add al and contents pointed by di.
 - Move al to result matrix
 - Increment si,di,bl
- Terminate the program

PROGRAM:

PROGRAM	COMMENTS
<pre> mov ax,data mov ds,ax mov cl,row1 mov dl,row2 cmp cl,dl jne last mov cl,col1 mov dl,col2 cmp cl,dl jne last mov al,row2 mul cl mov cx,ax mov si, offset matrix1 mov di, offset matrix2 mov bx, offset result </pre>	<p>Load data segment to ds</p> <p>Load row1 value to cl Load row2 value to dl Compare cl and dl Jump to last if not equal</p> <p>Load col1 value to cl Load col2 value to dl Compare cl and dl Jump to last if not equal</p> <p>Load row2 value to al Multiply al with cl Load value of ax to cx Load offset of matrix1 to si Load offset of matrix2 to di Load offset of result to bx</p>
<pre> Here: mov al, [si] add al, [di] mov [bx], al inc si inc di inc bx loop here </pre>	<p>cx register indicates the loop count</p> <p>Load contents pointed by si to al Add all with contents pointed by di Load al to result matrix Increment si Increment di Increment bx</p>
<pre> last: mov ah,4ch int 21h </pre>	<p>Terminate the program</p>

UNASSEMBLED CODE:

The screenshot displays a DOSBox environment with two main windows. The left window is the DOSBox command prompt, showing the execution of a program named 5A.EXE. The right window is the 5a.ASM Notepad editor, showing the assembly code for the program.

DOSBox Command Prompt:

```
DOSBox 0.74-3; CPU speed: 3000 cycles, Frameskip 0, Program: DEBUG
Microsoft Object Linker V2.01 (Large)
(C) Copyright 1982, 1983 by Microsoft Inc.

Run File [5A.EXE]:
List File [NUL.MAP]:
Libraries [.LIB]:
Warning: No STACK segment

There was 1 error detected.

E:\>DEBUG 5A.EXE
-U
076E:0100 B86A07      MOV     AX,076A
076E:0103 8ED8        MOV     DS,AX
076E:0105 8A0E0000     MOV     CL,[0000]
076E:0109 8A160100     MOV     DL,[0001]
076E:010D 3BD1        CMP     CL,DL
076E:010F 7528        JNZ     0139
076E:0111 8A0E0200     MOV     CL,[0002]
076E:0115 8A160300     MOV     DL,[0003]
076E:0119 3BD1        CMP     CL,DL
076E:011B 751C        JNZ     0139
076E:011D A00100     MOV     AL,[0001]
-
```

5a.ASM Notepad Editor:

```
File Edit Format View Help
assume cs:code,ds:data
data segment
    row1 db 02h
    row2 db 02h
    col1 db 04h
    col2 db 04h
    org 0010h
    matrix1 db 00h,11h,22h,33h,44h,55h,66h,77h
    org 0020h
    matrix2 db 77h,66h,55h,44h,33h,22h,11h,00h
    org 0030h
    result db 8 DUP(0)
data ends
code segment
    org 0100h
start: mov ax,data
       mov ds,ax
       mov cl,row1
       mov dl,row2
       cmp cl,dl
       jne last
       mov cl,col1
       mov dl,col2
       cmp cl,dl
       jne last
       mov al,row2
       mul cl
       mov cx,ax
       mov si, offset matrix1
       mov di, offset matrix2
       mov bx, offset result
```

SAMPLE INPUT/OUTPUT:

The screenshot shows a DOSBox window titled "DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Program: DEBUG". The main window displays assembly instructions being executed by the debugger. The instructions are as follows:

```

076E:011B 751C      JNZ     0139
076E:011D A00100     MOV     AL,[0001]

-D 076A:0000
076A:0000 02 02 04 04 00 00 00 00-00 00 00 00 00 00 00 00 ...
076A:0010 00 11 22 33 44 55 66 77-00 00 00 00 00 00 00 00 ...
076A:0020 77 66 55 44 33 22 11 00-00 00 00 00 00 00 00 00 wfUD
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 02 02 04 04 00 00 00 00-00 00 00 00 00 00 00 00 ...
076A:0010 00 11 22 33 44 55 66 77-00 00 00 00 00 00 00 00 ...
076A:0020 77 66 55 44 33 22 11 00-00 00 00 00 00 00 00 00 wfUD
076A:0030 77 77 77 77 77 77 77 77-00 00 00 00 00 00 00 00 wwwuu
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 ....


```

A secondary window titled "5A.ASM - Notepad" shows the source code corresponding to the assembly instructions displayed in the DOSBox window:

```

File Edit Format View Help

result db 8 DUP(0)

data ends

code segment

org 0100h
start: mov ax,data
       mov ds,ax
       mov cl,row1
       mov dl,row2
       cmp cl,dl
       jne last
       mov cl,col1
       mov dl,col2
       cmp cl,dl
       jne last
       mov al,row2
       mul cl
       mov cx,ax
       mov si, offset matrix1
       mov di, offset matrix2
       mov bx, offset result
here:  mov al,[si]
       add al,[di]
       mov [bx],al
       inc si
       inc di
       inc bl
       loop here
last:  mov ah,4ch
       int 21h

code ends
end start

Ln 1, Col 1

```

RESULT:

Thus matrix addition has been achieved.

B. AIM:

Program for matrix subtraction.

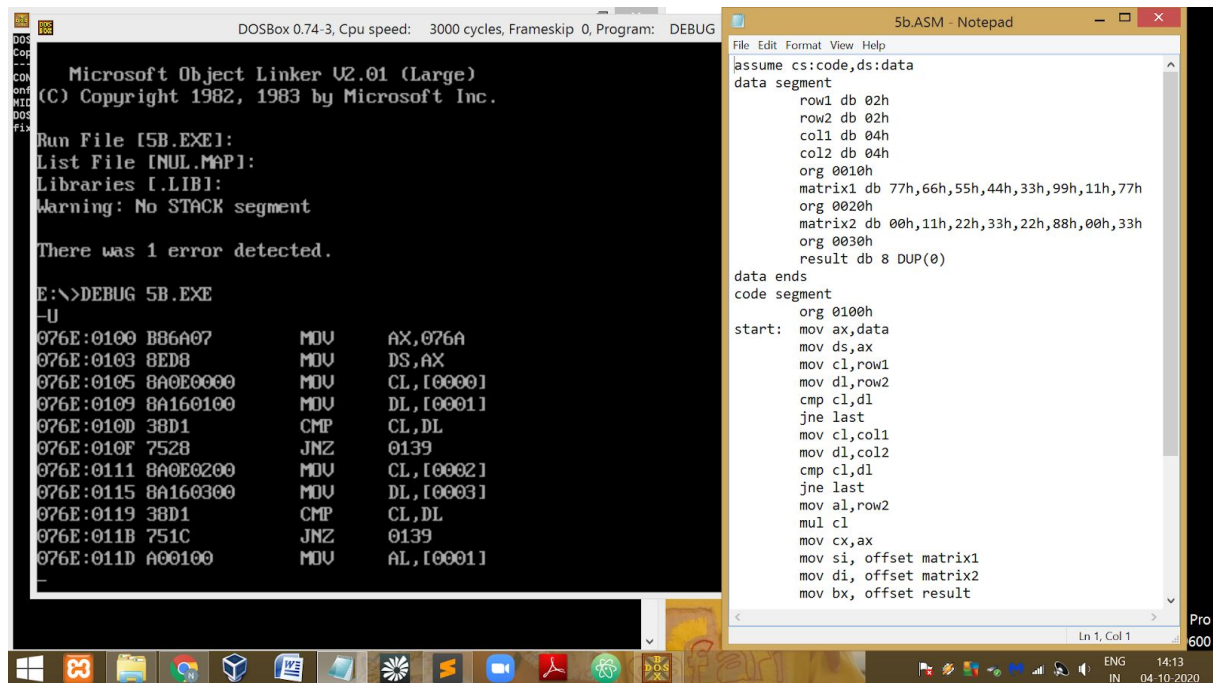
ALGORITHM:

- Initialize the data segment.
- Move data segment address to ds
- Load row1 to cl, row2 to dl.
- Compare cl and dl and terminate if not equal.
- Load col1 to cl, col2 to dl.
- Compare cl and dl and terminate if not equal.
- Move row2 to al.
- Multiply al with cl and move ax to cx.
- Move offset of matrix1 to si, matrix2 to di, result to bx
- Loop here:
 - Move contents pointed by si to al and subtract al and contents pointed by di from al.
 - Move al to result matrix
 - Increment si,di,bl
- Terminate the program

PROGRAM:

PROGRAM	COMMENTS
<pre> mov ax,data mov ds,ax mov cl,row1 mov dl,row2 cmp cl,dl jne last mov cl,col1 mov dl,col2 cmp cl,dl jne last mov al,row2 mul cl mov cx,ax mov si, offset matrix1 mov di, offset matrix2 mov bx, offset result </pre>	<pre> Load data segment to ds Load row1 value to cl Load row2 value to dl Compare cl and dl Jump to last if not equal Load col1 value to cl Load col2 value to dl Compare cl and dl Jump to last if not equal Load row2 value to al Multiply al with cl Load value of ax to cx Load offset of matrix1 to si Load offset of matrix2 to di Load offset of result to bx </pre>
<pre> Here: mov al, [si] add al, [di] mov [bx], al inc si inc di inc bl loop here </pre>	<pre> cx register indicates the loop count Load contents pointed by si to al Add all with contents pointed by di Load al to result matrix Increment si Increment di Increment bl </pre>
<pre> last: mov ah,4ch int 21h </pre>	<pre> Terminate the program </pre>

UNASSEMBLED CODE:



The screenshot shows a DOSBox window on the left and a Notepad window on the right. The DOSBox window displays the output of the Microsoft Object Linker V2.01, including the command 'Run File [5B.EXE]', the file 'NUL.MAP', and a warning about no STACK segment. It also shows the assembly code for 5B.EXE, starting with 'MOV AX,076A' and 'MOV DS,AX'. The Notepad window shows the assembly code for 5B.ASM, which includes data definitions for 'row1', 'row2', 'col1', 'col2', 'matrix1', 'matrix2', and 'result', and a code segment starting with 'start: mov ax,data' and 'mov ds,ax'.

```
Microsoft Object Linker V2.01 (Large)
(C) Copyright 1982, 1983 by Microsoft Inc.

Run File [5B.EXE]:
List File [NUL.MAP]:
Libraries [.LIB]:
Warning: No STACK segment

There was 1 error detected.

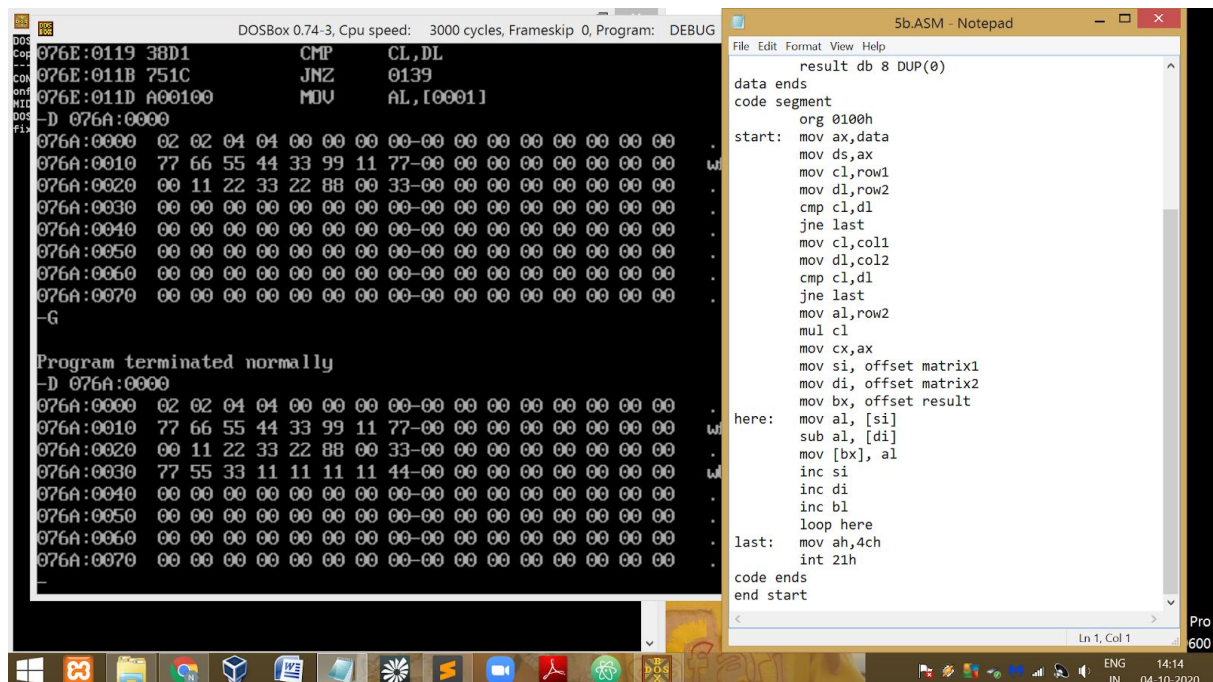
E:\>DEBUG 5B.EXE
-U
076E:0100 B86A07      MOV     AX,076A
076E:0103 8ED8        MOV     DS,AX
076E:0105 8A0E0000     MOV     CL,[0000]
076E:0109 8A160100     MOV     DL,[0001]
076E:010D 3BD1        CMP     CL,DL
076E:010F 7528        JNZ     0139
076E:0111 8A0E0200     MOV     CL,[0002]
076E:0115 8A160300     MOV     DL,[0003]
076E:0119 3BD1        CMP     CL,DL
076E:011B 751C        JNZ     0139
076E:011D A00100     MOV     AL,[0001]

076A:0000 02 02 04 04 00 00 00 00-00 00 00 00 00 00 00 00 .
076A:0010 77 66 55 44 33 99 11 77-00 00 00 00 00 00 00 00 .
076A:0020 00 11 22 33 22 88 00 33-00 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 .

Program terminated normally
-D 076A:0000
076A:0000 02 02 04 04 00 00 00 00-00 00 00 00 00 00 00 00 .
076A:0010 77 66 55 44 33 99 11 77-00 00 00 00 00 00 00 00 .
076A:0020 00 11 22 33 22 88 00 33-00 00 00 00 00 00 00 00 .
076A:0030 77 55 33 11 11 11 11 44-00 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 .

5B.ASM - Notepad
File Edit Format View Help
assume cs:code,ds:data
data segment
    row1 db 02h
    row2 db 02h
    col1 db 04h
    col2 db 04h
    org 0010h
    matrix1 db 77h,66h,55h,44h,33h,99h,11h,77h
    org 0020h
    matrix2 db 00h,11h,22h,33h,22h,88h,00h,33h
    org 0030h
    result db 8 DUP(0)
data ends
code segment
    org 0100h
start: mov ax,data
        mov ds,ax
        mov cl,row1
        mov dl,row2
        cmp cl,d1
        jne last
        mov cl,col1
        mov dl,col2
        cmp cl,d1
        jne last
        mov al,row2
        mul cl
        mov cx,ax
        mov si, offset matrix1
        mov di, offset matrix2
        mov bx, offset result
        here: mov al,[si]
                sub al,[di]
                mov [bx],al
                inc si
                inc di
                inc bx
                loop here
        last: mov ah,4ch
                int 21h
code ends
end start
```

SAMPLE INPUT/OUTPUT



The screenshot shows a DOSBox window on the left and a Notepad window on the right. The DOSBox window displays the assembly code for 5B.EXE, including the command 'Run File [5B.EXE]', the file 'NUL.MAP', and a warning about no STACK segment. It also shows the assembly code for 5B.EXE, starting with 'MOV AX,076A' and 'MOV DS,AX'. The Notepad window shows the assembly code for 5B.ASM, which includes data definitions for 'row1', 'row2', 'col1', 'col2', 'matrix1', 'matrix2', and 'result', and a code segment starting with 'start: mov ax,data' and 'mov ds,ax'.

```
Microsoft Object Linker V2.01 (Large)
(C) Copyright 1982, 1983 by Microsoft Inc.

Run File [5B.EXE]:
List File [NUL.MAP]:
Libraries [.LIB]:
Warning: No STACK segment

There was 1 error detected.

E:\>DEBUG 5B.EXE
-U
076E:0100 B86A07      MOV     AX,076A
076E:0103 8ED8        MOV     DS,AX
076E:0105 8A0E0000     MOV     CL,[0000]
076E:0109 8A160100     MOV     DL,[0001]
076E:010D 3BD1        CMP     CL,DL
076E:010F 7528        JNZ     0139
076E:0111 8A0E0200     MOV     CL,[0002]
076E:0115 8A160300     MOV     DL,[0003]
076E:0119 3BD1        CMP     CL,DL
076E:011B 751C        JNZ     0139
076E:011D A00100     MOV     AL,[0001]

076A:0000 02 02 04 04 00 00 00 00-00 00 00 00 00 00 00 00 .
076A:0010 77 66 55 44 33 99 11 77-00 00 00 00 00 00 00 00 .
076A:0020 00 11 22 33 22 88 00 33-00 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 .

Program terminated normally
-D 076A:0000
076A:0000 02 02 04 04 00 00 00 00-00 00 00 00 00 00 00 00 .
076A:0010 77 66 55 44 33 99 11 77-00 00 00 00 00 00 00 00 .
076A:0020 00 11 22 33 22 88 00 33-00 00 00 00 00 00 00 00 .
076A:0030 77 55 33 11 11 11 11 44-00 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 .

5B.ASM - Notepad
File Edit Format View Help
assume cs:code,ds:data
data segment
    row1 db 02h
    row2 db 02h
    col1 db 04h
    col2 db 04h
    org 0010h
    matrix1 db 77h,66h,55h,44h,33h,99h,11h,77h
    org 0020h
    matrix2 db 00h,11h,22h,33h,22h,88h,00h,33h
    org 0030h
    result db 8 DUP(0)
data ends
code segment
    org 0100h
start: mov ax,data
        mov ds,ax
        mov cl,row1
        mov dl,row2
        cmp cl,d1
        jne last
        mov cl,col1
        mov dl,col2
        cmp cl,d1
        jne last
        mov al,row2
        mul cl
        mov cx,ax
        mov si, offset matrix1
        mov di, offset matrix2
        mov bx, offset result
        here: mov al,[si]
                sub al,[di]
                mov [bx],al
                inc si
                inc di
                inc bx
                loop here
        last: mov ah,4ch
                int 21h
code ends
end start
```

RESULT:

Thus matrix subtraction has been achieved.