Experiment No 6: Sorting

Date: 04-10-2020 **NAME: Gayathri M**
REG.NO: 185001050

A. AIM:

Program for sorting in ascending order.

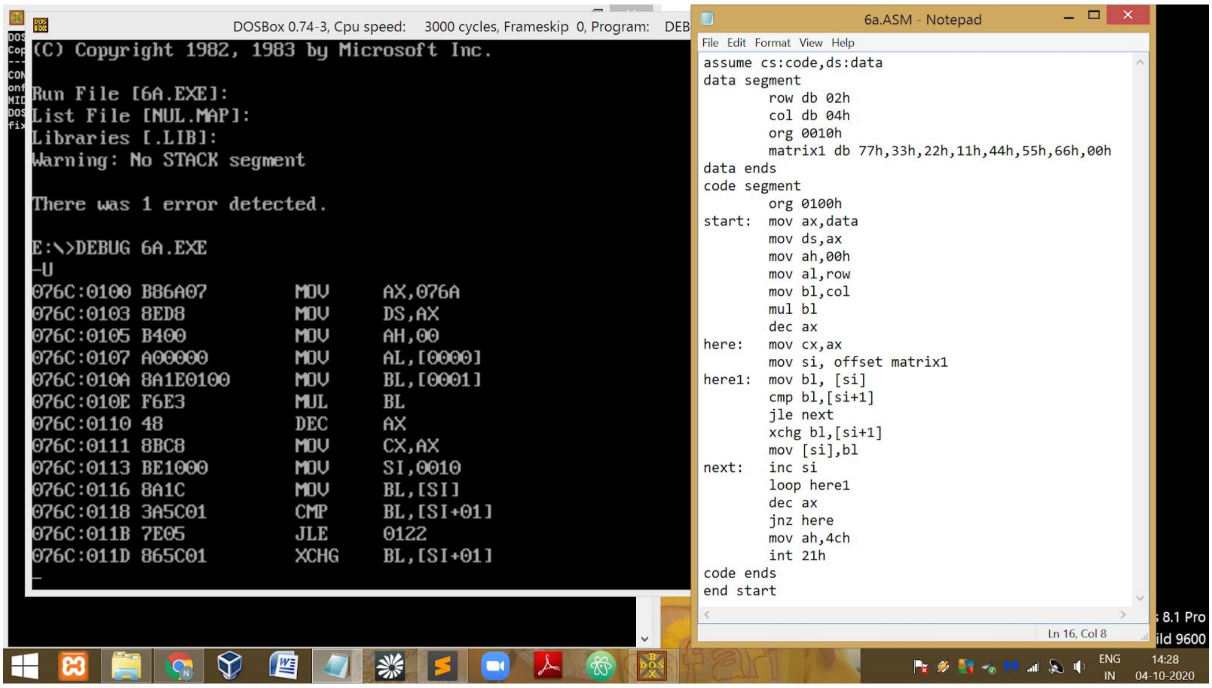
ALGORITHM:

- Initialize the data segment.
- Move data segment address to ds
- Initialize ah with 00h.
- Move row value to al and col value to bl.
- Multiply al with bl.
- Decrement ax.
- Here :
    - Move ax value to cx
    - Load offset of matrix1 to si
- Here1 :
    - Move contents pointed by si to bl
    - Compare contents pointed by si+1 with bl
    - If bl is less than or equal to [si+1] jump to next
    - Exchange values of bl and [si+1]
    - Move bl to matrix1
- next :
    - Increment si
    - Loop here1
    - Decrement ax
    - Jump to here if not equal to zero
    - Terminate the program

PROGRAM:

PROGRAM	**COMMENTS**
start:	
mov ax,data	Load data segment to ds
mov ds,ax	
mov ah,00h	Initialise ah with 00h
mov al,row	Move row value to ah
mov bl,col	Move col value to bl
mul bl	Multiply al with bl
dec ax	Decrement ax
Here:	
mov cx,ax	Move contents of ax to cx.
mov si, offset matrix1	Move offset of matrix1 to si.
here1:	
mov bl, [si]	Move contents pointed by si to bl
cmp bl,[si+1]	Move contents pointed by si+1 to bl
jle next	If bl is less than or equal to [si+1] jump to next
xchg bl,[si+1]	Exchange values of bl and [si+1]
mov [si],bl	Move bl to matrix1
next:	
inc si	Increment si
loop here1	Start loop here1
dec ax	Decrement ax
jnz here	Jump to here if not equal to 0
mov ah,4ch	Terminate the program
int 21h	

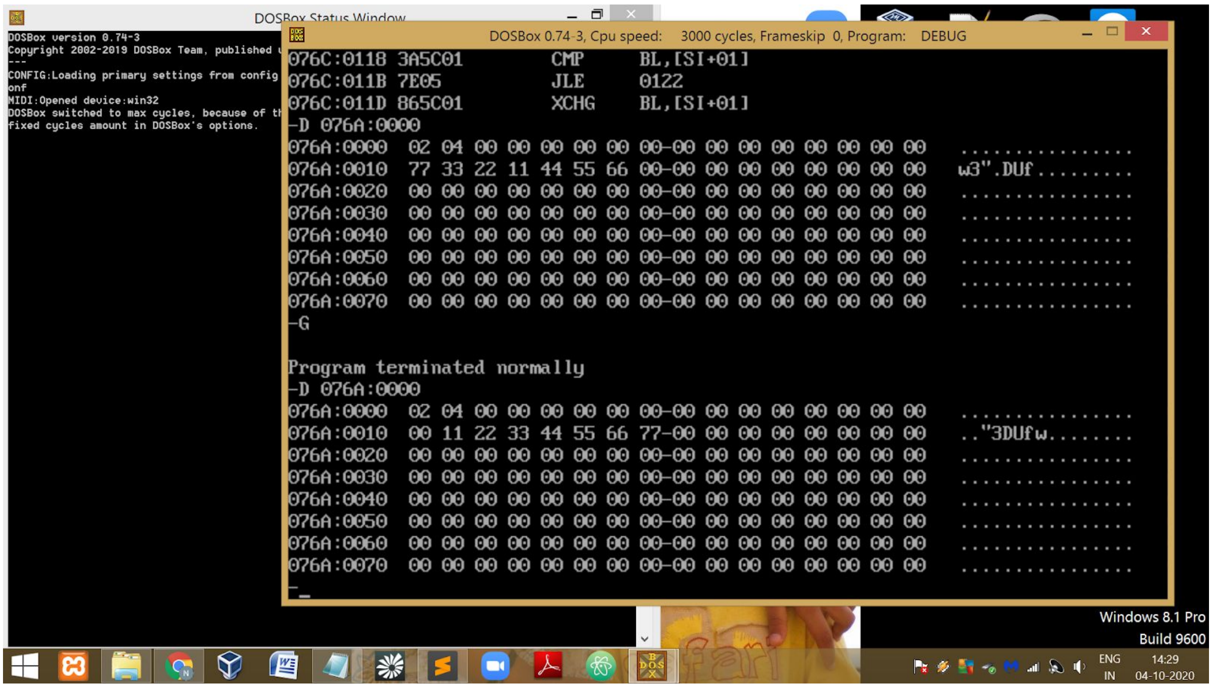
UNASSEMBLED CODE:


DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Program: DEB
6a.ASM - Notepad
File Edit Format View Help
Ln 16, Col 8
8.1 Pro
ild 9600
ENG
IN
14:28
04-10-2020

```
(C) Copyright 1982, 1983 by Microsoft Inc.

Run File [6A.EXE]:
List File [NUL.MAP]:
Libraries [.LIB]:
Warning: No STACK segment

There was 1 error detected.

E:\>DEBUG 6A.EXE
-U
076C:0100 B86A07        MOV     AX,076A
076C:0103 8ED8          MOV     DS,AX
076C:0105 B400          MOV     AH,00
076C:0107 A00000        MOV     AL,[0000]
076C:010A 8A1E0100      MOV     BL,[0001]
076C:010E F6E3          MUL     BL
076C:0110 48            DEC     AX
076C:0111 8BC8          MOV     CX,AX
076C:0113 BE1000        MOV     SI,0010
076C:0116 8A1C          MOV     BL,[SI]
076C:0118 3A5C01        CMP     BL,[SI+01]
076C:011B 7E05          JLE     0122
076C:011D 865C01        XCHG    BL,[SI+01]
-
```

```
assume cs:code,ds:data
data segment
        row db 02h
        col db 04h
        org 0010h
        matrix1 db 77h,33h,22h,11h,44h,55h,66h,00h
data ends
code segment
        org 0100h
start:  mov ax,data
        mov ds,ax
        mov ah,00h
        mov al,row
        mov bl,col
        mul bl
        dec ax
here:   mov cx,ax
        mov si, offset matrix1
here1:  mov bl, [si]
        cmp bl,[si+1]
        jle next
        xchg bl,[si+1]
        mov [si],bl
next:   inc si
        loop here1
        dec ax
        jnz here
        mov ah,4ch
        int 21h
code ends
end start
```

SAMPLE INPUT/OUTPUT:


DOSBox Status Window
DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Program: DEBUG
Windows 8.1 Pro
Build 9600
ENG
IN
14:29
04-10-2020

```
DOSBox version 0.74-3
Copyright 2002-2019 DOSBox Team, published u
---
CONFIG:Loading primary settings from config
onf
MIDI:Opened device:win32
DOSBox switched to max cycles, because of th
fixed cycles amount in DOSBox's options.
```

```
076C:0118 3A5C01        CMP     BL,[SI+01]
076C:011B 7E05          JLE     0122
076C:011D 865C01        XCHG    BL,[SI+01]
-D 076A:0000
076A:0000  02 04 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
076A:0010  77 33 22 11 44 55 66 00-00 00 00 00 00 00 00 00   w3".DUf.........
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  02 04 00 00 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   .."3DUfw........
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   ................
-
```

RESULT:

Thus sorting in ascending order is achieved.

B. AIM:

Program for sorting in descending order.

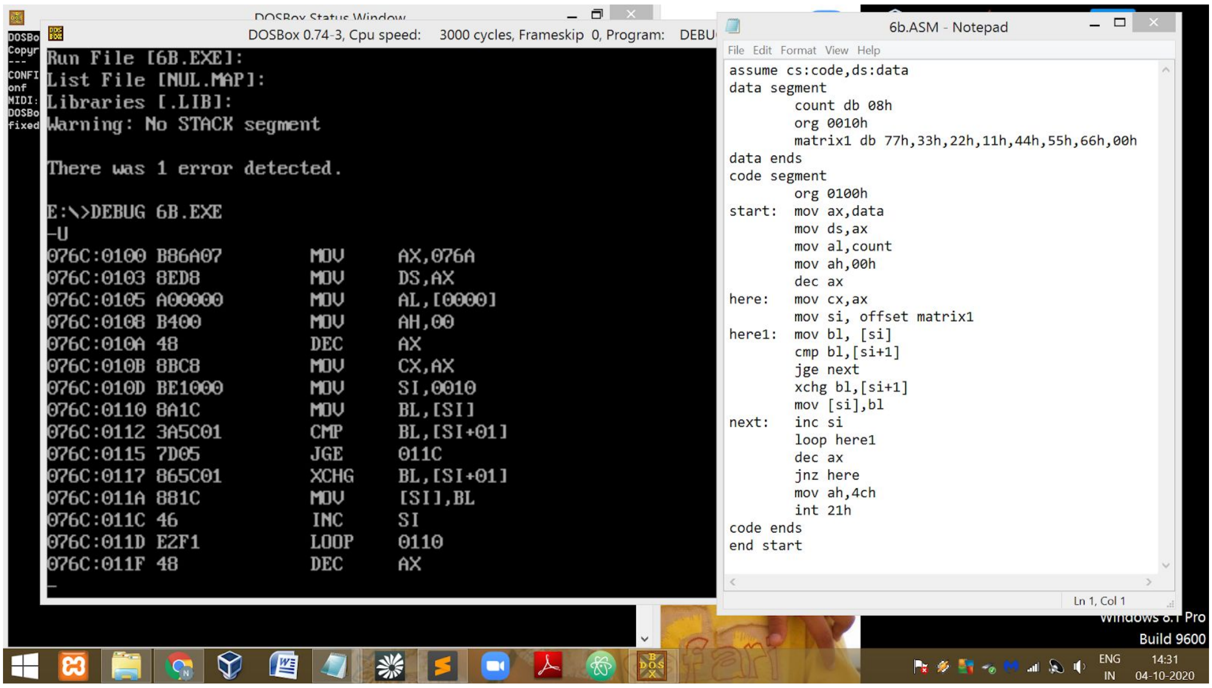
ALGORITHM:

- Initialize the data segment.
- Move data segment address to ds
- Load al with count value
- Initialize ah with 00h.
- Decrement ax.
- Here :
    - Move ax value to cx
    - Load offset of matrix1 to si
- Here1 :
    - Move contents pointed by si to bl
    - Compare contents pointed by si+1 with bl
    - If bl is greater than or equal to [si+1] jump to next
    - Exchange values of bl and [si+1]
    - Move bl to matrix1
- next :
    - Increment si
    - Loop here1
    - Decrement ax
    - Jump to here if not equal to zero
    - Terminate the program

PROGRAM:

PROGRAM	**COMMENTS**
start:	
mov ax,data	Load data segment to ds
mov ds,ax	
mov al,count	Load al with count.
mov ah,00h	Initialise ah with 00h
dec ax	Decrement ax
Here:	
mov cx,ax	Move contents of ax to cx.
mov si, offset matrix1	Move offset of matrix1 to si.
here1:	
mov bl, [si]	Move contents pointed by si to bl
cmp bl,[si+1]	Move contents pointed by si+1 to bl
jge next	If bl is greater than or equal to [si+1] jump to next
xchg bl,[si+1]	Exchange values of bl and [si+1]
mov [si],bl	Move bl to matrix1
next:	
inc si	Increment si
loop here1	Start loop here1
dec ax	Decrement ax
jnz here	Jump to here if not equal to 0
mov ah,4ch	Terminate the program
int 21h	

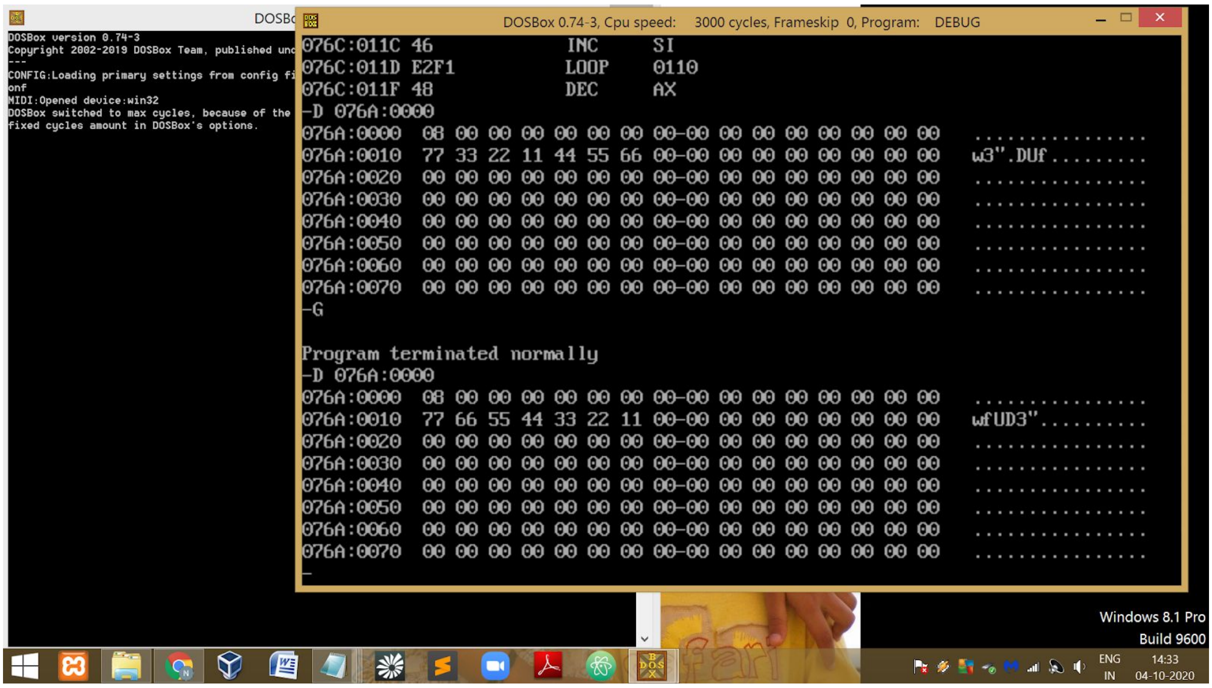
UNASSEMBLED CODE:


DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Program: DEBU
6b.ASM - Notepad
File Edit Format View Help
Ln 1, Col 1
Windows 8.1 Pro
Build 9600
ENG
IN
14:31
04-10-2020

```
Run File [6B.EXE]:
List File [NUL.MAP]:
Libraries [.LIB]:
Warning: No STACK segment

There was 1 error detected.

E:\>DEBUG 6B.EXE
-U
076C:0100 B86A07        MOV     AX,076A
076C:0103 8ED8          MOV     DS,AX
076C:0105 A00000        MOV     AL,[0000]
076C:0108 B400          MOV     AH,00
076C:010A 48            DEC     AX
076C:010B 8BC8          MOV     CX,AX
076C:010D BE1000        MOV     SI,0010
076C:0110 8A1C          MOV     BL,[SI]
076C:0112 3A5C01        CMP     BL,[SI+01]
076C:0115 7D05          JGE     011C
076C:0117 865C01        XCHG    BL,[SI+01]
076C:011A 881C          MOV     [SI],BL
076C:011C 46            INC     SI
076C:011D E2F1          LOOP    0110
076C:011F 48            DEC     AX
-
```

```
assume cs:code,ds:data
data segment
        count db 08h
        org 0010h
        matrix1 db 77h,33h,22h,11h,44h,55h,66h,00h
data ends
code segment
        org 0100h
start:  mov ax,data
        mov ds,ax
        mov al,count
        mov ah,00h
        dec ax
here:   mov cx,ax
        mov si, offset matrix1
here1:  mov bl, [si]
        cmp bl,[si+1]
        jge next
        xchg bl,[si+1]
        mov [si],bl
next:   inc si
        loop here1
        dec ax
        jnz here
        mov ah,4ch
        int 21h
code ends
end start
```

SAMPLE INPUT/OUTPUT


DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Program: DEBUG
Windows 8.1 Pro
Build 9600
ENG
IN
14:33
04-10-2020

```
DOSBox version 0.74-3
Copyright 2002-2019 DOSBox Team, published unc
---
CONFIG:Loading primary settings from config fi
onf
MIDI:Opened device:win32
DOSBox switched to max cycles, because of the
fixed cycles amount in DOSBox's options.
```

```
076C:011C 46            INC     SI
076C:011D E2F1          LOOP    0110
076C:011F 48            DEC     AX
-D 076A:0000
076A:0000  08 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
076A:0010  77 33 22 11 44 55 66 00-00 00 00 00 00 00 00 00   w3".DUf.........
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  08 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
076A:0010  77 66 55 44 33 22 11 00-00 00 00 00 00 00 00 00   wfUD3"..........
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   ................
-
```

RESULT:

Thus sorting in descending order is achieved.