Exp No: 6 `**Date:**08 /10/2020

Sorting

Name: Swetha Saseendran

Reg No: 185001183

Aim:

To program and execute the sorting of 8 bit N values in ascending and descending order in 8086 microprocessor using DOSBOX.

Programs:

(i) SORTING IN ASCENDING ORDER

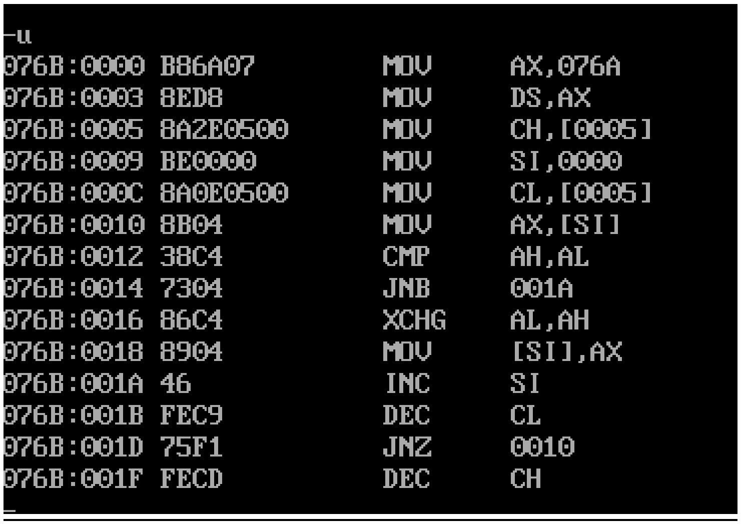
Algorithm:

- Program is set to run from any specified memory position.
- Load data from arr to register AX.
- Compare the digits in arr move the smaller to front and larger to back.
- Use the instruction XCHG to mov of e between the digits
- Move the digits until zero flag becomes zero and length of arr becomes zero
- Terminate the program.

Program	Comments
;To sort a set of numbers in an arr in ascending order	
DATA SEGMENT	
arr DB 05H, 04H, 03H, 02H, 01H	Array with 05, 04, 03, 02, 01 as input
arrlen DB 04H	Array length as 04
DATA ENDS	
ASSUME CS:CODE,DS:DATA	
Code SEGMENT	
START: MOV AX,DATA	Address of data segment moved to ax

MOV DS,AX MOV CH, arrlen ;outer loop iteration OUTER: MOV SI, offset(arr) MOV CL, arrlen ; INNER: MOV AX, [SI] CMP AH, AL JNC SKIP XCHG AL, AH MOV [SI], AX SKIP: INC SI DEC CL JNZ INNER DEC CH JNZ OUTER MOV AH,4CH INT 21H Code ENDS END START END	Address of ax moved to ax Value of arrlen moved to ch Starting pointer of arr Inner loop iteration (reinitialize) Jump if no carry to SKIP AH AL is stored together Decrease inner loop Decease outer loop Terminate the program

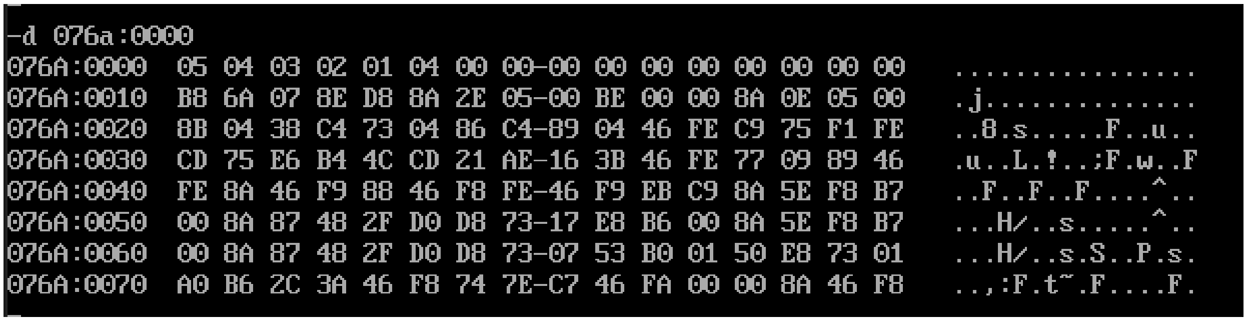
Unassembled Code:



```
-u
076B:0000 B86A07        MOV     AX,076A
076B:0003 8ED8          MOV     DS,AX
076B:0005 8A2E0500      MOV     CH,[0005]
076B:0009 BE0000        MOV     SI,0000
076B:000C 8A0E0500      MOV     CL,[0005]
076B:0010 8B04          MOV     AX,[SI]
076B:0012 38C4          CMP     AH,AL
076B:0014 7304          JNB     001A
076B:0016 86C4          XCHG    AL,AH
076B:0018 8904          MOV     [SI],AX
076B:001A 46            INC     SI
076B:001B FEC9          DEC     CL
076B:001D 75F1          JNZ     0010
076B:001F FECD          DEC     CH
-
```

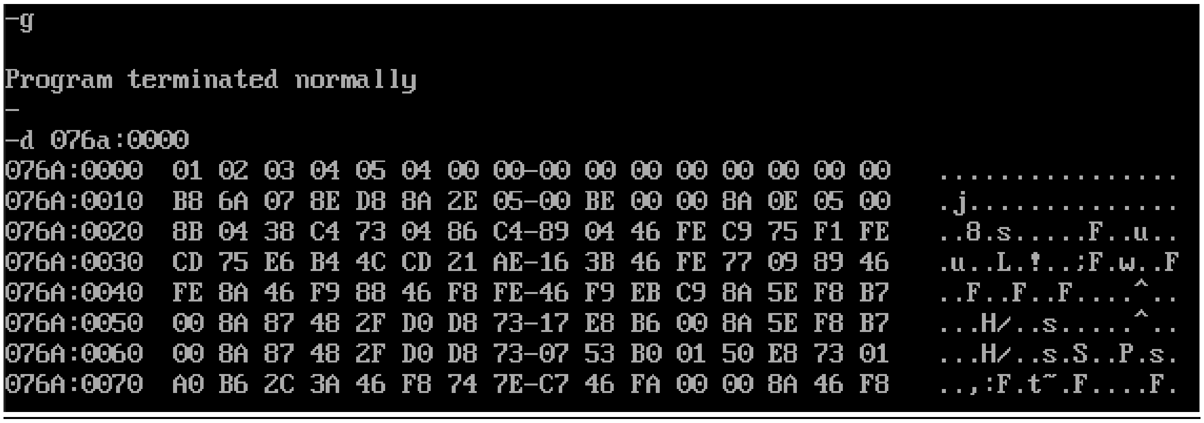
Snapshot of sample input and output:

INPUT:



```
-d 076a:0000
076A:0000  05 04 03 02 01 04 00 00-00 00 00 00 00 00 00 00   ................
076A:0010  B8 6A 07 8E D8 8A 2E 05-00 BE 00 00 8A 0E 05 00   .j..............
076A:0020  8B 04 38 C4 73 04 86 C4-89 04 46 FE C9 75 F1 FE   ..8.s.....F..u..
076A:0030  CD 75 E6 B4 4C CD 21 AE-16 3B 46 FE 77 09 89 46   .u..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.
```

OUTPUT:



```
-g

Program terminated normally
-
-d 076a:0000
076A:0000  01 02 03 04 05 04 00 00-00 00 00 00 00 00 00 00   ................
076A:0010  B8 6A 07 8E D8 8A 2E 05-00 BE 00 00 8A 0E 05 00   .j..............
076A:0020  8B 04 38 C4 73 04 86 C4-89 04 46 FE C9 75 F1 FE   ..8.s.....F..u..
076A:0030  CD 75 E6 B4 4C CD 21 AE-16 3B 46 FE 77 09 89 46   .u..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.
```

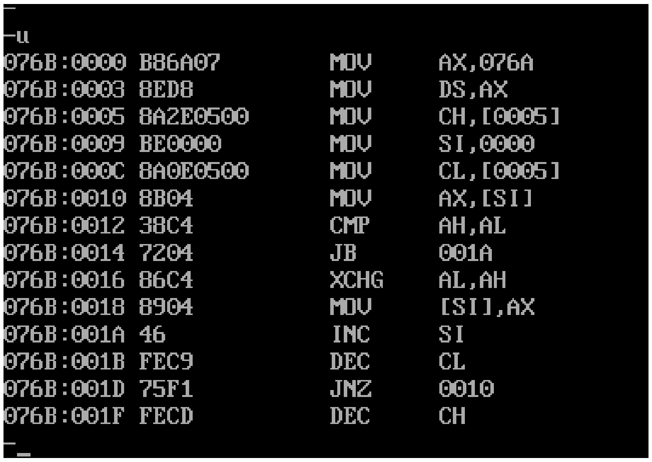
(ii) SORTING IN DESCENDING ORDER

Algorithm:

- Program is set to run from any specified memory position.
- Load data from arr to register AX.
- Compare the digits in arr move the larger digit to front and smaller digit to back of arr.
- Use the instruction XCHG to move between the digits
- Move the digits until zero flag becomes zero and length of arr becomes zero
- Terminate the program.

Program	Comments
;To sort a set of numbers in an arr in descending order	
DATA SEGMENT	
arr DB 01H, 02H, 03H, 04H, 05H	Array with 01, 02, 03, 04, 05 as input
arrlen DB 04H	Array length as 04
DATA ENDS	
ASSUME CS:CODE,DS:DATA	
Code SEGMENT	
START: MOV AX,DATA	Address of data segment moved to ax
MOV DS,AX	Address of ax moved to ax
MOV CH, arrlen	Value of arrlen moved to ch
OUTER: MOV SI, offset(arr)	Starting pointer of arr
MOV CL, arrlen	Inner loop iteration (reinitialize)
INNER: MOV AX, [SI]	
CMP AH, AL	
JC SKIP	Jump if carry to SKIP
XCHG AL, AH	
MOV [SI], AX	AH AL is stored together
SKIP: INC SI	
DEC CL	Decrease inner loop
JNZ INNER	
DEC CH	Decease outer loop
JNZ OUTER	
MOV AH,4CH	
INT 21H	
Code ENDS	Terminate the program
END START	
END	

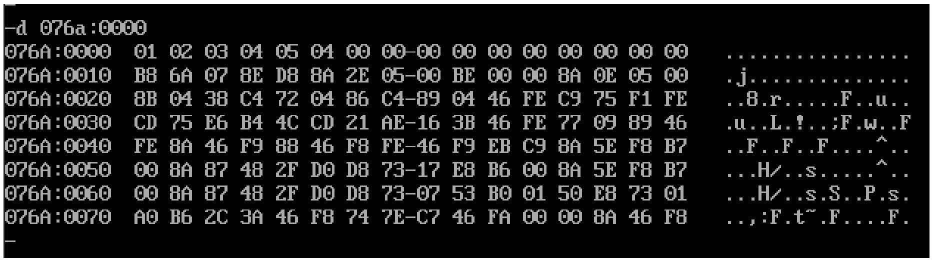
Unassembled Code:



```
-
-u
076B:0000 B86A07        MOV     AX,076A
076B:0003 8ED8          MOV     DS,AX
076B:0005 8A2E0500      MOV     CH,[0005]
076B:0009 BE0000        MOV     SI,0000
076B:000C 8A0E0500      MOV     CL,[0005]
076B:0010 8B04          MOV     AX,[SI]
076B:0012 38C4          CMP     AH,AL
076B:0014 7204          JB      001A
076B:0016 86C4          XCHG    AL,AH
076B:0018 8904          MOV     [SI],AX
076B:001A 46            INC     SI
076B:001B FEC9          DEC     CL
076B:001D 75F1          JNZ     0010
076B:001F FECD          DEC     CH
-_
```

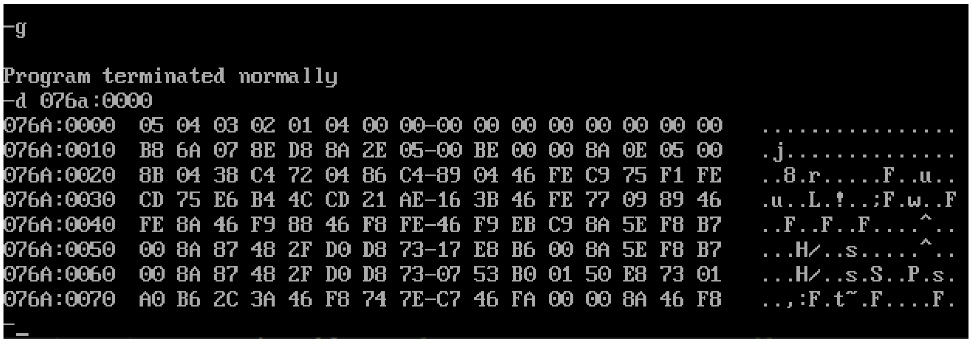
Snapshot of sample input and output:

INPUT:



```
-
-d 076a:0000
076A:0000  01 02 03 04 05 04 00 00-00 00 00 00 00 00 00 00   ................
076A:0010  B8 6A 07 8E D8 8A 2E 05-00 BE 00 00 8A 0E 05 00   .j..............
076A:0020  8B 04 38 C4 72 04 86 C4-89 04 46 FE C9 75 F1 FE   ..8.r.....F..u..
076A:0030  CD 75 E6 B4 4C CD 21 AE-16 3B 46 FE 77 09 89 46   .u..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.
-
```

OUTPUT:



```
-g

Program terminated normally
-d 076a:0000
076A:0000  05 04 03 02 01 04 00 00-00 00 00 00 00 00 00 00   ................
076A:0010  B8 6A 07 8E D8 8A 2E 05-00 BE 00 00 8A 0E 05 00   .j..............
076A:0020  8B 04 38 C4 72 04 86 C4-89 04 46 FE C9 75 F1 FE   ..8.r.....F..u..
076A:0030  CD 75 E6 B4 4C CD 21 AE-16 3B 46 FE 77 09 89 46   .u..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:

Therefore, the ascending and descending sorting are performed and verified using MASM.