

Date: 27-01-2021 Exp. 3 Sorting in Ascending and Descending Order



Aim:

To write an assembly language program to arrange numbers in ascending and descending order and hence print the largest and smallest in the array using bubble sort.

Tool Used:

Assembler - MASM 611

Algorithm:

Step 1: Start

Step 2: Store the Fifteen numbers in an array

Step 3: Compare each number with their adjacent numbers and swap according to the requirement of getting ascending or descending respectively.

Step 4: Observe the output and verify the results

Step 5: End

Repeat this procedure for complete series ($n-1$) times. After $n-1$ iterations you will get the largest number at the end of the series and smallest at the beginning of the series.

Program:

ASSUME CS: CODE, DS:DATA

DATA SEGMENT

ARRAY1 DB 54H, 67H, 47H, 11H, 51H, 61H, 45H, 76H, 74H, 12H, 15H, 16H, 20H,
11H, 97H

MIN DB 00H

ARRAY2 DB 54H, 67H, 47H, 11H, 51H, 61H, 45H, 76H, 74H, 12H, 15H, 16H, 20H,
11H, 97H

MAX DB 00H

DATA ENDS

CODE SEGMENT

START:

MOV AX, DATA

MOV DS, AX

XOR AX, AX

MOV CH, 0EH

L1: MOV CL, 0EH

LEA SI, ARRAY1

L2: MOV AL, [SI]

MOV BL, [SI+1]

CMP AL, BL

JC DOWN

MOV DL, [SI+1]

XCHG [SI], DL

MOV[SI+1], DL

DOWN: INC SI

DEC CL

JNZ L2

DEC CH

JNZ L1

LEA SI, ARRAY1

MOV AL, [SI]

MOV MIN, AL

L3: MOV CL, 0EH

LEA SI, ARRAY2

L4: MOV AL, [SI]

MOV BL, [SI+1]

CMP AL, BL

JNC UP

MOV DL, [SI+1]

XCHG [SI], DL

MOV [SI+1], DL

UP: INC SI

DEC CL

JNZ L4

DEC CH

JNZ L3

LEA SI, ARRAY2

MOV AL, [SI]

MOV MAX, AL

HLT

CODE ENDS

END START

Sample Input:

54,67,47,11,51,61,45,76,74,12,15,16,20,11,97

Sample Output:

Ascending: 11,11,12,15,16,20,45,47,51,54,61,67,74,76,97

Descending: 97,76,74,67,61,54,51,47,45,20,16,15,12,11,11

Minimum element: 11

Maximum element: 97

Manual Verification:

Bubble Sort Visualization

Run | Reset | List size: 15 ▾

Your values: 54,67,47,11,51,61,45,76,74,12,15,16,20,11,97

208 / 208

<< < > >>

Done sorting!

11	11	12	15	16	20	45	47	51	54	61	67	74	76	97
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14

```

1. void bubblesort(Comparable* A[], int n) {
2.     for (int i = 0; i < n-1; i++) // Insert i'th record
3.         for (int j = 0; j < n-i; j++)
4.             if (*A[j] > *A[j+1])
5.                 swap(A, j, j+1);
6. }

```

Register/ Memory Contents for I/O:

-U		
0766:005B	F4	HLT
0766:005C	0406	ADD AL,06
0766:005E	43	INC BX
0766:005F	98	CBW
0766:0060	92	XCHG DX,AX
0766:0061	92	XCHG DX,AX
0766:0062	2202	AND AL,[BP+SI]
0766:0064	04AD	ADD AL,AD
0766:0066	42	INC DX
0766:0067	04A5	ADD AL,A5
0766:0069	52	PUSH DX
0766:006A	4A	DEC DX
0766:006B	04C0	ADD AL,C0
0766:006D	04EA	ADD AL,EA
0766:006F	04A9	ADD AL,A9
0766:0071	0429	ADD AL,29
0766:0073	A5	MOVSW
0766:0074	9D	POPF
0766:0075	0409	ADD AL,09
0766:0077	1CFE	SBB AL,FE
0766:0079	08F2	OR DL,DH

Snapshot of the Output:

```
-g 005B

AX=0097  BX=0011  CX=0000  DX=0076  SP=0000  BP=0000  SI=0010  DI=0000
DS=0764  ES=0754  SS=0763  CS=0766  IP=005B    NV UP EI PL ZR NA PE NC
0766:005B F4          HLT
-D 0764:0000 001F
0764:0000  11 11 12 15 16 20 45 47-51 54 61 67 74 76 97 11   .... EGQTagtv..
0764:0010  97 76 74 67 61 54 51 47-45 20 16 15 12 11 11 97  .vtgaTQGE ....
-
```

Result:

The elements in the respective arrays were arranged in ascending and descending order using bubble sort and smallest and largest numbers were printed.