

Bubble sort

.MODEL SMALL

DISPLAY MACRO MSG

LEA DX, MSG

MOV AH, 09H

INT 21H

ENDM

.DATA

LIST DB 02H, 01H, 34H, 0F4H, 09H, 05H

NUMBER EQU \$-LIST

MSG1 DB 0DH, 0AH, "1 >> SORT IN ASCENDING ORDER"

MSG2 DB 0DH, 0AH, "2 >> SORT IN DESCENDING ORDER"

MSG3 DB 0DH, 0AH, "3 >> EXIT"

MSG4 DB 0DH, 0AH, "ENTER YOUR CHOICE :: \$"

MSG5 DB 0DH, 0AH, "INVALID CHOICE ENTERED...\$"

.CODE

START : MOV AX, @DATA

MOV DS, AX

LEA SI, LIST

MOV CH, NUMBER-1 ; CL STORES THE NUMBER OF ELEMENTS IN LIST

DISPLAY MSG1 ; DISPLAY THE MENU...

DISPLAY MSG2

DISPLAY MSG3

DISPLAY MSG4

MOV AH, 01H

INT 21H

SUB AL, 30H

CMP AL, 01H ; INPUT=1? SORT IN ASCENDING ORDER

JE ASCSORT

CMP AL, 02H ; INPUT=2? SORT IN DESCENDING ORDER

<

Close

Ln 1, Col 1

100%

Windows (CRLF)

UTF-8



Search



11:36

02-12-2020



JE DESSORT

CMP AL, 03H ; INPUT=3? EXIT

JE FINAL

DISPLAY MSG5

JMP FINAL

ASCSORT:MOV BL, 00H

AGAIN: MOV SI, OFFSET LIST

MOV CL, 00H ; J VALUE

MOV BH, CH

SUB BH, BL ; N-1-i

NPASS: CMP CL, BH

JNC NEXT

MOV AL, [SI]

MOV BP, 01H

CMP AL, DS:[BP][SI]

JC _NOPE

XCHG AL, [SI+1]

XCHG [SI], AL

_NOPE : INC CL

INC SI

JMP NPASS

NEXT: INC BL

CMP BL, CH

JC AGAIN

JMP FINAL

DESSORT:MOV BL, 00H

AGAIN1: MOV SI, OFFSET LIST

MOV CL, 00H ; J VALUE

<

>

Ln 1, Col 1

100%

Windows (CRLF)

UTF-8



Search



11:36
02-12-2020



CMP BL, CH
JC AGAIN
JMP FINAL

DESSORT:MOV BL, 00H
AGAIN1: MOV SI, OFFSET LIST
MOV CL, 00H ; J VALUE
MOV BH, CH
SUB BH, BL ; N-1-i
NPASS1: CMP CL, BH
JNC NEXT
MOV AL, [SI]
MOV BP, 01H
CMP AL, DS: [BP][SI]
JNC _NOPE1

XCHG AL, [SI+1]
XCHG [SI], AL
_NOPE1: INC CL
INC SI
JMP NPASS1
NEXT1: INC BL
CMP BL, CH
JC AGAIN1
FINAL: MOV AH, 4CH
INT 21H
END START

Bubble sort.

model small

data

n dw 5

a dw 05, 07, 04, 03, 06

code

mov ax, @data

mov DS, ax

mov cx, n

dec cx

outloop: mov ch, cx

mov SI, 00h

inloop: mov al, a[SI]

inc SI

cmp al, a[SI]

JC noexch

xchg al, a[SI]

mov a[SI-1], al

noexch: Dec ch

Inc inloop

Dec cx

JNC outloop

mov ah, 4ch

Int 21h

end.

The screenshot shows the DOS DEBUG program in assembly mode. The main window displays a list of assembly instructions with their addresses, hex values, and mnemonics. The instruction at address 002F is highlighted. On the right, the register window shows the current values of various registers. At the bottom, a hex dump displays the memory contents starting from address 0000.

Address	Hex Value	Mnemonic	Comment
0020	B44C	MOV	AH,4C
0021	CD21	INT	21
002F	0085	ADD	Byte Ptr [DI],AL
0031	050204	ADD	AX,0402
0034	03062009	ADD	AX,Word Ptr [0020]
0038	D89782	CALL	0002
003B	C706E6012000	MOV	Word Ptr [01E6],0020
0041	06	PUSH	ES
0042	BE86E401	MOV	ES,Word Ptr [01E4]
0046	268B162C00	MOV	DX,Word Ptr ES:[002C]
004B	D07D82	CALL	0009
004E	1F52	MOV	HI,52
0050	B9FF7F	MOV	CX,7FFF
0053	B02000	MOV	AX,0020
0056	D87982	CALL	0002
0059	07	POP	ES

Register	Value
AX	4C25
CX	0000
DX	0000
SI	0000
DI	0000
BP	0000
SP	0000
DS	052D
ES	051A
SS	0529
CS	052A
IP	002F
FL	0007

Address	Hex Value
0000	05 05 07 04 01 06 20 00 00 97 02 C7 06 E6 01 20
0010	00 06 8E 06 E4 01 26 8B 16 2C 00 E8 7D 02 B7 92

At the bottom, the command line shows: (F0=Trace) (F10=Step) (F5=Go) (F6=Window) (F3=Display)