**SSN College of Engineering Department of Computer Science and Engineering**

**III year - UCS1512 – Microprocessors Lab**

**Sorting**

**Exp No:** 06

**Name:** Rahul Ram M

**Register Number:** 185001121

**Date:** 04/10/2020

**a) Sorting in ascending order:**

**Aim:**

Design 8086 program for sorting in ascending order.

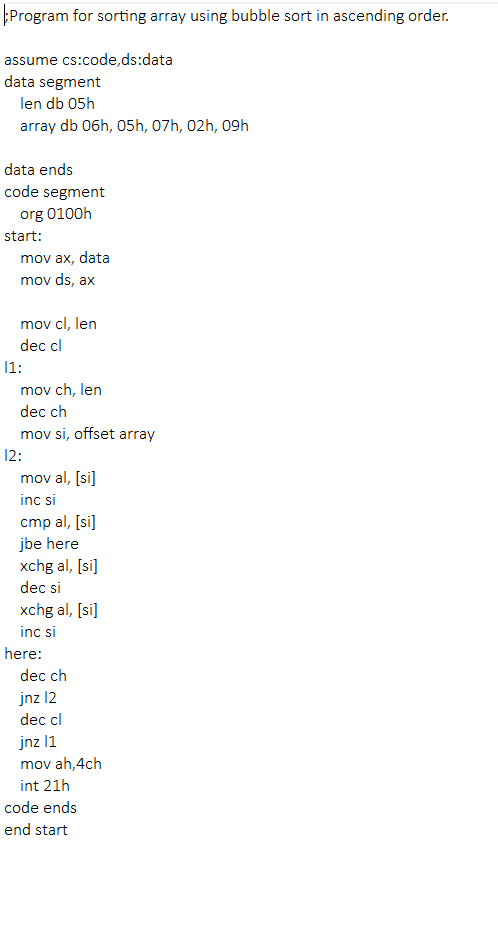
**Procedure for executing MASM:**

1. Run Dosbox and mount your masm folder to a drive in dosbox.
2. Goto the mounted drive.
3. Save the 8086 program with extension .asm in the same folder using command “edit”
4. After creating the file, assemble it using the command “masm filename.asm”
5. Link the file using the command “link filename.obj;”
6. Use debug command with filename.exe to execute and analyse the memory contents, “debug filename.exe”.
7. In debug, command “u” will display the unassembled code.
8. Use command “d segment:offset” to see the content of memory locations starting from segment:offset address.
9. To change the value in memory, use the command “e segment:offset”
10. Verify the memory contents to ensure the updates (using command “d”).
11. . Execute using the command “g” and check the outputs.
12. “q” to exit from debug and “exit” to exit from command prompt and to close the Dosbox.

**Algorithm:**

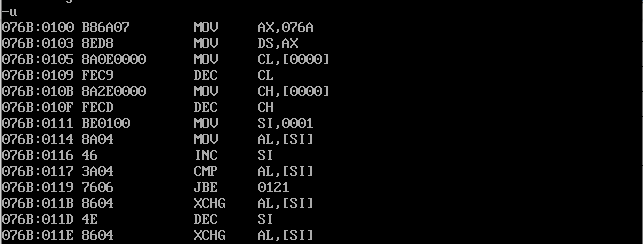
1. Move the starting address of data segment to AX register and move the data from AX register to DS register.
2. Move LEN to CH register and decrement, it.
3. L1: Move LEN to CL register and decrement, it.
4. Move the starting address of the array to SI register.
5. L2: Move the value stored in SI’s register to AL register and increment SI register.
6. Compare the contents at AL register and at SI register’s location using CMP.
7. Jump to HERE if the first value is less than or equal to second value.
8. Exchange the value at AL and SI register and decrement SI.
9. Now exchange values at AL and new SI’s location and increment SI.
10. HERE: Decrement CH. If CH’s value is not equal to zero, then jump to L2.
11. Decrement CL. If CL’s value is not equal to zero, then jump to L1.
12. INT 21H means invoke the interrupt identified by the hexadecimal number 21. In MS-DOS, invoking interrupt 21h while AH = 4Ch causes the current process to terminate and uses the value of register AL as the exit code of the process.

**Program:**

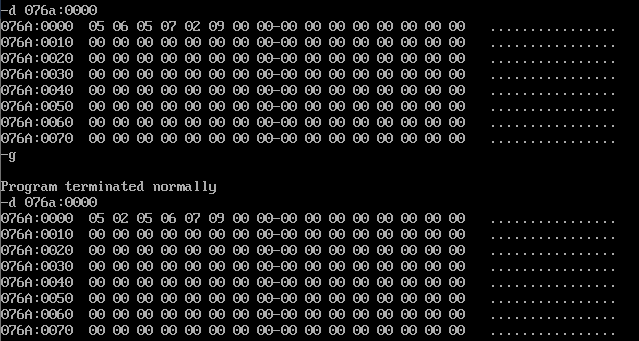


|  |  |  |
| --- | --- | --- |
|  | **Program** | **Comments** |
| START: | ORG 0100H | Memory instruction starts from 0100H. |
| MOV AX, DATA  MOV DS, AX | Transferring the data from DATA to AX register and  from AX register to DS register. |
| MOV CL, LEN | CL <- LEN |
| DEC CL | CL <- CL - 1 |
| L1: | MOV CH, LEN | CH <- LEN |
| DEC CH | CH <- CH -1 |
| MOV SI, OFFSET ARRAY | SI <- OFFSET ARRAY |
| L2: | MOV AL, [SI] | AL <- [SI] |
| INC SI | SI <- SI + 1 |
| CMP AL, [SI] | Compare AL and {SI] |
| JBE HERE | Jump to HERE, if AL is below/equal to [SI] |
| XCHG AL, [SI] | Exchange contents at AL and [SI] |
| DEC SI | SI <- SI - 1 |
| XCHG AL, [SI] | Exchange contents at AL and [SI] |
| INC SI | SI <- SI + 1 |
| HERE: | DEC CH | CH <- CH - 1 |
| JNZ L2 | Jump to L2 if CH is zero. |
| DEC CL | CL <- CL - 1 |
| JNZ L1 | Jump to L2 if CL is zero. |
| MOV AH, 4CH  INT 21H | Terminates the program. |

**Snapshot of sample input and output:**



**Sorting the array in ascending order (06h, 05h, 07h, 02h, 09h):**



**Result:**

Thus the 8086 program for sorting in ascending order is executed successfully in DOS-BOX.

**b) Sorting in descending order:**

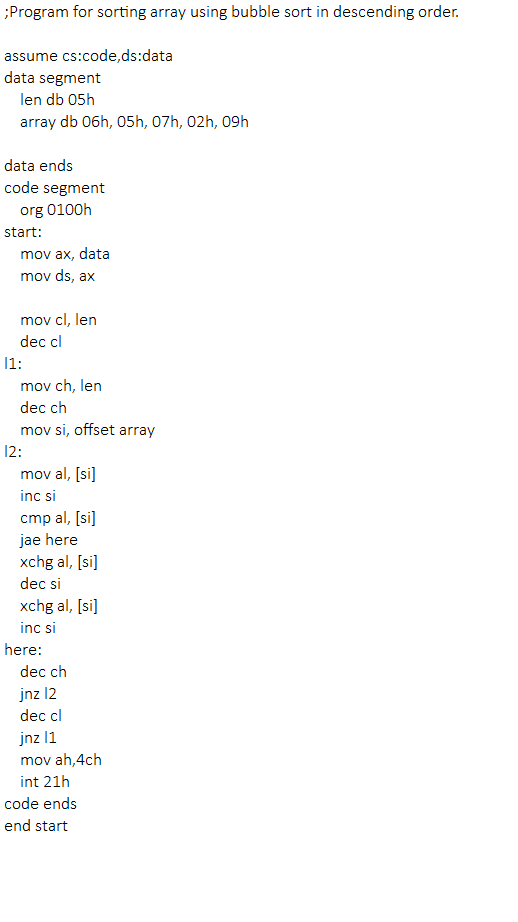
**Aim:**

Design 8086 program for Sorting in descending order.

**Algorithm:**

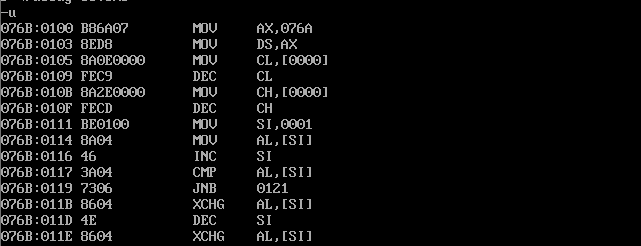
1. Move the starting address of data segment to AX register and move the data from AX register to DS register.
2. Move LEN to CH register and decrement, it.
3. L1: Move LEN to CL register and decrement, it.
4. Move the starting address of the array to SI register.
5. L2: Move the value stored in SI’s register to AL register and increment SI register.
6. Compare the contents at AL register and at SI register’s location using CMP.
7. Jump to HERE if the first value is greater than or equal to second value.
8. Exchange the value at AL and SI register and decrement SI.
9. Now exchange values at AL and new SI’s location and increment SI.
10. HERE: Decrement CH. If CH’s value is not equal to zero, then jump to L2.
11. Decrement CL. If CL’s value is not equal to zero, then jump to L1.
12. INT 21H means invoke the interrupt identified by the hexadecimal number 21. In MS-DOS, invoking interrupt 21h while AH = 4Ch causes the current process to terminate and uses the value of register AL as the exit code of the process.

**Program:**

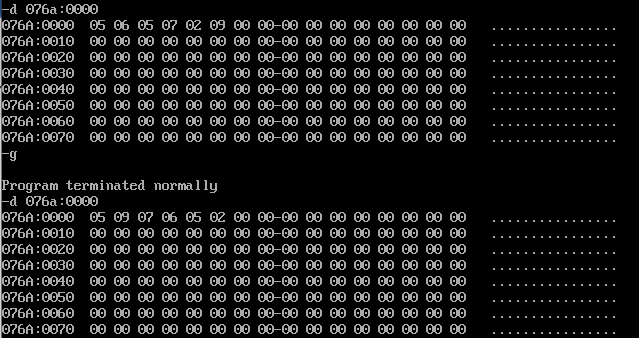


|  |  |  |
| --- | --- | --- |
|  | **Program** | **Comments** |
| START: | ORG 0100H | Memory instruction starts from 0100H. |
| MOV AX, DATA  MOV DS, AX | Transferring the data from DATA to AX register and  from AX register to DS register. |
| MOV CL, LEN | CL <- LEN |
| DEC CL | CL <- CL - 1 |
| L1: | MOV CH, LEN | CH <- LEN |
| DEC CH | CH <- CH -1 |
| MOV SI, OFFSET ARRAY | SI <- OFFSET ARRAY |
| L2: | MOV AL, [SI] | AL <- [SI] |
| INC SI | SI <- SI + 1 |
| CMP AL, [SI] | Compare AL and {SI] |
| JAE HERE | Jump to HERE, if AL is above/equal to [SI] |
| XCHG AL, [SI] | Exchange contents at AL and [SI] |
| DEC SI | SI <- SI - 1 |
| XCHG AL, [SI] | Exchange contents at AL and [SI] |
| INC SI | SI <- SI + 1 |
| HERE: | DEC CH | CH <- CH - 1 |
| JNZ L2 | Jump to L2 if CH is zero. |
| DEC CL | CL <- CL - 1 |
| JNZ L1 | Jump to L2 if CL is zero. |
| MOV AH, 4CH  INT 21H | Terminates the program. |

**Snapshot of sample input and output:**



**Sorting the array in descending order (06h, 05h, 07h, 02h, 09h):**



**Result:**

Thus the 8086 program for sorting in descending order is executed successfully in

DOS-BOX.