

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Programme | : | **BTech. CSE Core** | Semester | : | **Win 2021-22** |
| Course | : | **Microprocessor and Interfacing** | Code | : | **CSE2006** |
| Faculty | : | **Dr. Florence Gnana Poovathy J** | Slot | : | **L15+L16** |
| Name | : | **Hariket Sukesh Kumar Sheth** | Register No. | : | **20BCE1975** |

**Experiment 3:**Sorting the elements of an Array in Ascending and Descending order

**1**

|  |  |  |  |
| --- | --- | --- | --- |
| Date: 02-02-2022 | Exp. 03 | Sorting Array |  |

**Sort Array Elements in Ascending Order**

**Aim:** To Perform Sorting (Ascending Order) for the elements of the Array

**Tool Used:** Assembler – MASM611

**Algorithm:**

**Step 1:** First of all, mount the c drive using the command: **mount c** **c:\masm611\bin**

**Step 2:** After pressing **enter,** type **c:** and press enter.

**Step 3:** Now give a command, **ascend.asm** for writing/editing the code and the write the code.

**Step 4:** A pop window appears; there we have to write out code(instructions) following the logic given below.

1. Start the Data Segment.
2. Declare Array Numbers and Store five Hexadecimal Numbers.
3. Data Segment Ends.
4. Start Code Segment.
5. Assume Ds is Data and Cs is Code.
6. Move Data into AX.
7. Move AX into DS.
8. Move 04H into CH.
9. Start LOOP2 and Mov 04H into CL.
10. Load the Effective Address (LEA) into SI from Numbers.
11. Start LOOP1 o Move [SI] into AL and [SI+1] into BL.
12. Compare BL and AL.
13. JC Down.

**2**

1. Move [SI+1] into DL.
2. Exchange [SI] and DL.
3. Move DL into [SI+1].
4. Start Down.
5. Increment SI.
6. Decrement CL.
7. JNZ LOOP1.
8. Decrement CH.
9. JNZ LOOP2.

HLT and End of Code Segment

**Step 5:** Now give a command, **masm ascend.asm** for running the code. The object file is created.

**Step 6:** Now give a command, **link ascend.obj** to link the object file to library file present in the bin folder.

**Step 7:** Press **ENTER** four times.

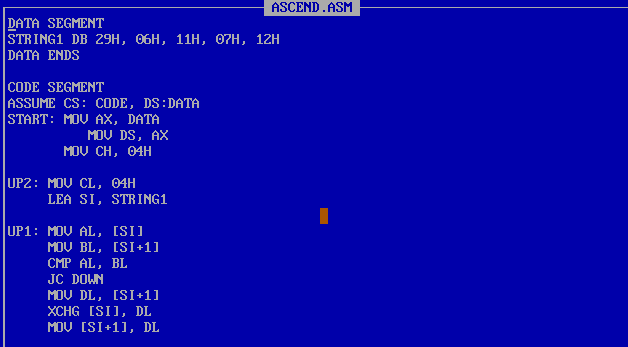
**Step 8:** Write debug **ascend.exe**

**-u**

**-g** (followed by the **address of HLT or INT** to view the values in registers).

**-d** (followed by the address of the Data Segment and index of the Array 0 to 4)

**Program:**

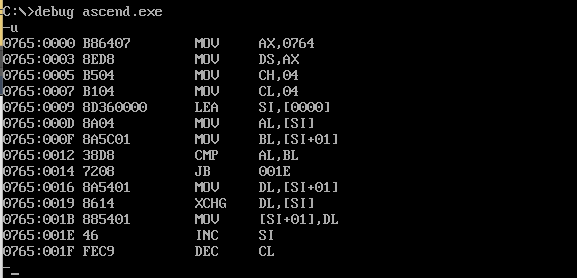
****



**3**

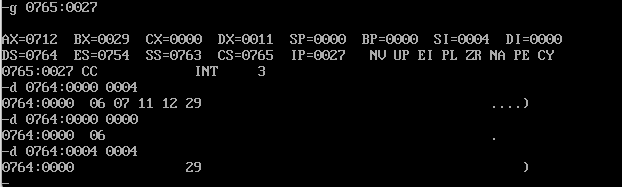
|  |  |
| --- | --- |
| **Sample Input:** | **Sample Output:** |
| String1: 29H, 06H, 11H, 07H, 12H | String1: 06H, 07H, 11H, 12H, 29H  Maximum Element: 29H  Minimum Element: 06H |

**Register / Memory Contents for I/O:**





**Output:**



**4**

|  |  |  |  |
| --- | --- | --- | --- |
| Date: 02-02-2022 | Exp. 03 | Sorting Array |  |

**Sort Array Elements in Descending Order**

**Aim:** To Perform Sorting (Descending Order) for the elements of the Array

**Tool Used:** Assembler – MASM611

**Algorithm:**

**Step 1:** First of all, mount the c drive using the command: **mount c** **c:\masm611\bin**

**Step 2:** After pressing **enter,** type **c:** and press enter.

**Step 3:** Now give a command, **descend.asm** for writing/editing the code and the write the code.

**Step 4:** A pop window appears; there we have to write out code(instructions) following the logic given below.

1. Start the Data Segment.
2. Declare Array Numbers and Store five Hexadecimal Numbers
3. Data Segment Ends.
4. Start Code Segment.
5. Assume Ds is Data and Cs is Code.
6. Move Data into AX.
7. Move AX into DS.
8. Move 04H into CH.
9. Start LOOP2 and Mov 04H into CL.
10. Load the Effective Address (LEA) into SI from Numbers.
11. Start LOOP1.
12. Move [SI] into AL and [SI+1] into BL.

**5**

1. Compare BL and AL.
2. JNC Down.
3. Move [SI+1] into DL.
4. Exchange [SI] and DL.
5. Move DL into [SI+1].
6. Start Down.
7. Increment SI.
8. Decrement CL.
9. JNZ LOOP1.
10. Decrement CH.
11. JNZ LOOP2.
12. HLT and End of Code Segment.
13. End of Program

**Step 5:** Now give a command, **masm descend.asm** for running the code. The object file is created.

**Step 6:** Now give a command, **link descend.obj** to link the object file to library file present in the bin folder.

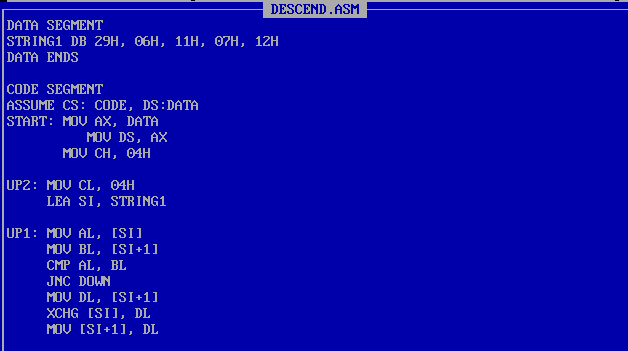
**Step 7:** Press **ENTER** four times.

**Step 8:** Write debug **descend.exe**

**-u**

**-g** (followed by the **address of HLT or INT** to view the values in registers).

**-d** (followed by the address of the Data Segment and index of the Array 0 to 4)

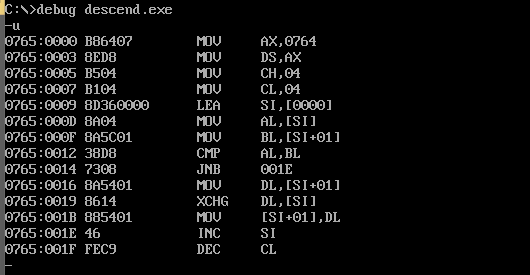
**Program:**

**6**



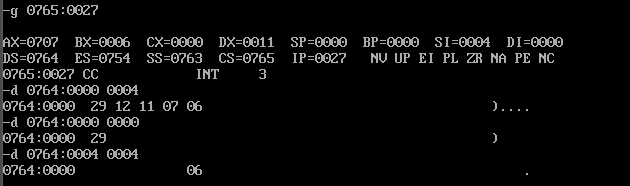
|  |  |
| --- | --- |
| **Sample Input:** | **Sample Output:** |
| String1: 29H, 06H, 11H, 07H, 12H | String1: 29H, 12H, 11H, 07H, 06H  Last Element: 06H  First Element: 29H |

**Register / Memory Contents for I/O:**



****

**Output:**



**7**