Exp No: 6 Matrix sorting Date: 12-10-2020 **Name: Anirudh H**

**Register Number: 185001019**

1. Aim:

To write assembly program to do following sorting operations:

Sorting in ascending order

Sorting in descending order

1. Procedure for executing MASM:
   1. Open DOS BOX simulation environment
   2. Mount the directory where the assembler code is present
   3. Move to mounted directory
   4. Assemble code using “masm<filename>.asm” command
   5. Link code the corresponding Obj file using “link <filename>.obj” command
   6. Use semicolon to avoid renaming the .exe file
   7. Debug created .exe file using “debug <filename>.exe” command
   8. Once inside the debug menu type
      * u: see code
      * d: see the data segment
      * g: run code
      * g: (2nd time) exit
   9. Exit
2. Algorithm
   1. Ascending

* Start
* Decrement ax.
* Here loop starts.
* Load contents of memory location ax in register cx.
* Load contents of memory location offset list in register si.
* Here1 loop starts.
* Load contents of memory location [si] in register bl.
* Compare bl and [si+1].
* Jump to next loop if lesser or equal to, else continue .
* Exchange bl and [si+1].
* Load contents of memory location bl in register [si].
* Next loop starts.
* Increment si.
* Loop to here1.
* Decrement ax.
* End
  1. Desending
     + Start
     + Decrement ax.
     + Here loop starts.
     + Load contents of memory location ax in register cx..
     + Load contents of memory location offset list in register si.
     + Here1 loop starts.
     + Load contents of memory location [si] in register bl.
     + Compare bl and [si+1].
     + Jump to next loop if greater or equal to, else continue.
     + Exchange bl and [si+1].
     + Load contents of memory location bl in register [si].
     + Next loop starts.
     + Increment si.
     + Loop to here1.
     + Decrement ax.
     + End

1. Program:
   1. Ascending

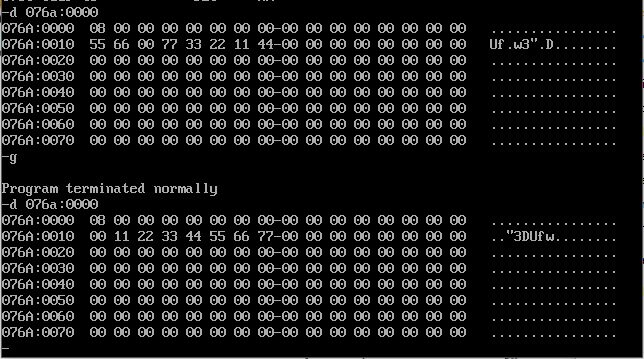
|  |  |
| --- | --- |
| Program | Comments |
| assume cs:code,ds:data,es:extra | Initializing the code, data and extra  segments to assembler |
| data segment | Data segment |
| count db 08h | count is declared and initialized to 02h |
| org 0010h | list data segment starting from address  range 0010h |
| assume cs:code,ds:data,es:extra | Initializing the code, data and extra  segments to assembler |
| data segment | Data segment |
| count db 08h | count is declared and initialized to 02h |
| org 0010h | list data segment starting from address  range 0010h |
| list db 55h,66h,00h,77h,33h,22h,11h,44h | list is declared and initialized to 55h,66h,00h,77h,33h,22h,11h,44h |
| data ends |  |
| code segment | Code segment |
| org 0100h | assemble the code starting from address range 0100h |
| start: mov ax,data | Transferring the data from memory location data to ax |
| mov ds,ax | Transferring the data from memory location ax to ds |
| mov al,count | Transferring the data from memory location count to al |
| mov ah,00h | Transferring the data 00h to ah |
| dec ax | Decrement ax |
| Here: mov cx,ax | Here loop  Transferring the data from memory location ax to cx |
| mov si, offset list | Transferring the data from memory location offset list to si |
| here1: mov bl, [si] | Here1 loop  Transferring the data from memory location [si] to bl |
| cmp bl,[si+1] | Compare bl and [si+1] |
| jle next | Jump to next loop if lesser or equal to, else continue |
| xchg bl,[si+1] | Exchange bl and [si+1] |
| mov [si],bl | Transferring the data from memory location bl to [si] |
| next: inc si | Next loop  Increment si |
| loop here1 | Loop to here1 |
| dec ax | Decrement ax |
| jnz here | If zero flag cleared jump to here else continue |
| mov ah,4ch | Transferring the termination code 4ch to ah |
| int 21h | Termination |
| code ends | Code ends |

* 1. Desending

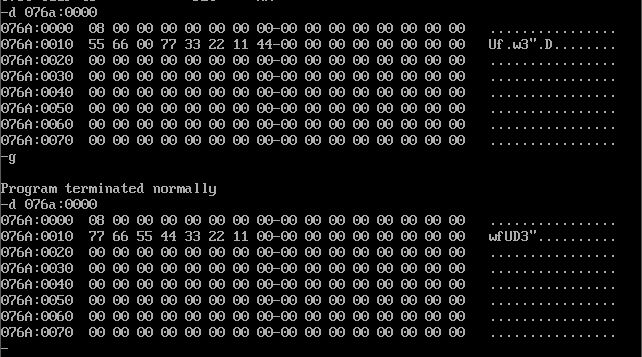
|  |  |
| --- | --- |
| Program | Comments |
| assume cs:code,ds:data,es:extra | Initializing the code, data and extra  segments to assembler |
| data segment | Data segment |
| count db 08h | count is declared and initialized to 02h |
| org 0010h | list data segment starting from address  range 0010h |
| list db 55h,66h,00h,77h,33h,22h,11h,44h | list is declared and initialized to  55h,66h,00h,77h,33h,22h,11h,44h |
| data ends |  |
| org 0100h | assemble the code starting from address range 0100h |
| start: mov ax,data | Transferring the data from memory location data to ax |
| mov ds,ax | Transferring the data from memory location ax to ds |
| mov al,count | Transferring the data from memory location count to al |
| mov ah,00h | Transferring the data 00h to ah |
| dec ax | Decrement ax |
| Here: mov cx,ax | Here loop  Transferring the data from memory location ax to cx |
| mov si, offset list | Transferring the data from memory location offset list to si |
| here1: mov bl, [si] | Here1 loop  Transferring the data from memory location [si] to bl |
| cmp bl,[si+1] | Compare bl and [si+1] |
| jge next | Jump to next loop if greater or equal to, else continue |
| xchg bl,[si+1] | Exchange bl and [si+1] |
| mov [si],bl | Transferring the data from memory location bl to [si] |
| next: inc si | Next loop  Increment si |
| loop here1 | Loop to here1 |
| dec ax | Decrement ax |
| jnz here | If zero flag cleared jump to here else continue |
| mov ah,4ch | Transferring the termination code 4ch to ah |
| int 21h | Termination |
| code ends | Code ends |
| end start |  |

Snapshot of sample input and output:

* 1. Ascending



* 1. Subtraction



1. Result

The assembly program for matrix sort is written and executed