Exp No: 5 Matrix Operations Date: 12-10-2020 **Name: Anirudh H**

**Register Number: 185001019**

1. Aim:

To write assembly program to do following:

Matrix addition

Matrix subtraction

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. Matrix addition

* Start
* Compare row1 and row2
* If not equal exit else continue
* Load contents of memory location col1 in register cl
* Load contents of memory location col2 in register dl
* Compare col1 and col2
* If not equal exit else continue
* Load contents of memory location row2 in register al
* Multiply col1 and row2
* Load contents of memory location ax in register cx
* End
  1. Matrix Subtraction
     + Start
     + Compare row1 and row2
     + If not equal exit else continue
     + Load contents of memory location col1 in register cl
     + Load contents of memory location col2 in register dl
     + Compare col1 and col2
     + If not equal exit else continue
     + Load contents of memory location row2 in register al
     + Multiply col1 and row2
     + Load contents of memory location ax in register cx
     + End

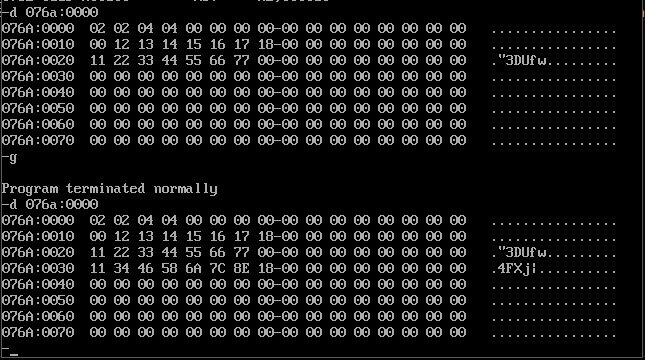
1. Program:
   1. Addition

|  |  |
| --- | --- |
| Program | Comments |
| assume cs:code,ds:data,es:extra | Initializing the code, data and extra segments to assembler |
| data segment | Data segment |
| row1 db 02h | row1 is declared and initialized to 02h |
| row2 db 02h | row2 is declared and initialized to 02h |
| col1 db 04h | col1 is declared and initialized to 04h |
| col2 db 04h | col2 is declared and initialized to 04h |
| org 0010h | Mat1 data segment starting from address range 0010h |
| mat1 db 00h,12h,13h,14h,15h,16h,17h,18h | mat1 is declared and initialized to 00h,12h,13h,14h,15h,16h,17h,18h |
| org 0020h | Mat2 data segment starting from address range 0020h |
| mat2 db 11h,22h,33h,44h,55h,66h,77h,00h | Mat2 is declared and initialized to 11h,22h,33h,44h,55h,66h,77h,00h |
| org 0030h | result data segment starting from address range 0030h |
| result db 8 DUP(0) | result is declared and initialized to empty |
| 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 cl,row1 | Transferring the data from memory location row1 to cl |
| mov dl,row2 | Transferring the data from memory location row2 to dl |
| cmp cl,dl | Compare cl and dl |
| mov cl,col1 | If not equal jump to last, else continue |
| mov dl,col2 | Transferring the data from memory location col1 to cl |
| cmp cl,dl | Transferring the data from memory location col2 to dl |
| jne last | Compare cl and dl |
| mov al,row2 | If not equal jump to last, else continue |
| mul cl | Transferring the data from memory location row2 to al |
| mov cx,ax | Multiply cl |
| mov si, offset mat1 | Transferring the data from memory  location ax to cx |
| mov di, offset mat2 | Transferring the data from memory location offset mat1 to si |
| mov bx, offset result | Transferring the data from memory location offset mat2 to di |
| here: mov al, [si] | Transferring the data from memory location offset result to bx |
| add al, [di] | Here loop starts  Transferring the data from memory location [si] to al |
| mov [bx], al | Add al => al+[di] |
| inc si | Transferring the data from memory  location [bx] to al |
| inc di | Increment si |
| inc bl | Increment di |
| loop here | Increment bl |
| last: mov ah,4ch | Loop continues |
| int 21h | Transferring the termination code 4ch to  ah |
| code ends | Termination |
| end start | Code ends |

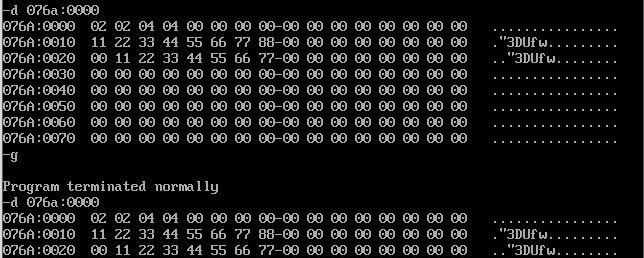
* 1. Subtraction

|  |  |
| --- | --- |
| Program | Comments |
| assume cs:code,ds:data,es:extra | Initializing the code, data and extra segments to assembler |
| data segment | Data segment |
| row1 db 02h | row1 is declared and initialized to 02h |
| row2 db 02h | row2 is declared and initialized to 02h |
| col1 db 04h | col1 is declared and initialized to 04h |
| org 0010h | col2 is declared and initialized to 04h |
| mat1 db 11h,22h,33h,44h,55h,66h,77h,88h | Mat1 data segment starting from address range 0010h |
| org 0020h | mat1 is declared and initialized to 11h,22h,33h,44h,55h,66h,77h,88h |
| mat2 db 00h,11h,22h,33h,44h,55h,66h,77h | Mat2 data segment starting from address range 0020h |
| org 0030h | Mat2 is declared and initialized to 00h,11h,22h,33h,44h,55h,66h,77h |
| result db 8 DUP(0) | result data segment starting from address range 0030h |
| data ends | result is declared and initialized to empty |
| code segment |  |
| org 0100h | Code segment |
| start: mov ax,data | assemble the code starting from address range 0100h |
| mov ds,ax | Transferring the data from memory location data to ax |
| mov cl,row1 | Transferring the data from memory location ax to ds |
| mov dl,row2 | Transferring the data from memory location row1 to cl |
| cmp cl,dl | Transferring the data from memory location row2 to dl |
| jne last | Compare cl and dl |
| mov cl,col1 | If not equal jump to last, else continue |
| mov dl,col2 | Transferring the data from memory location col1 to cl |
| cmp cl,dl | Transferring the data from memory location col2 to dl |
| jne last | Compare cl and dl |
| mov al,row2 | If not equal jump to last, else continue |
| mul cl | Transferring the data from memory location row2 to al |
| mov cx,ax | Multiply cl |
| mov si, offset mat1 | Transferring the data from memory  location ax to cx |
| mov bx, offset result | Transferring the data from memory location offset mat1 to si |
| here: mov al, [si] | Transferring the data from memory location offset mat2 to di |
| sub al, [di] | Transferring the data from memory location offset result to bx |
| mov [bx], al | Here loop starts  Transferring the data from memory location [si] to al |
| inc si | sub al => al+[di] |
| inc di | Transferring the data from memory location [bx] to al |
| inc bl | Increment si |
| loop here | Increment di |
| last: mov ah,4ch | Increment bl |
| int 21h | Loop continues |
| code ends | Transferring the termination code 4ch to ah |
| end start | Termination |
|  | Code ends |
|  |  |

1. Snapshot of sample input and output:
   1. Addition



* 1. Subtraction



1. Result

The assembly program for matrix subtraction is written and executed