Exp No: 3 **String Manipulations**

Date: 07/9/2020

Name: Gokhulnath T
Register Number: 185001051

Aim: To write assembly program to do following String Manipulations:

a) Moving a string of bytes
b) Comparing 2 strings of bytes
c) Searching a byte in a string
d) Moving a string without using string instructions

Procedure:

1. Install all the required file for executing MASM programs.(Masm, edit, link, debug etc..).
2. Write the assembly program in any editor before mounting the folder to the MASM.
3. Mount the folder that contains the assembly program with any name such as “d”.
    - mount d e:\masm
4. Create the object file of the assembly program using masm.
    - masm 16BITADD.asm
5. Use the link to create the executable file of the object file created from the above step.
    - Link 16BITADD.obj
6. Run the executable file using debug.
    - debug 16BITADD.exe
7. By un-assembling the program you can check the code segment of the program
    - u 076b:0100
8. To check the data memory segment, you can use the memory option to view the data stored.
    - d 076a:0000
9. To enter your own values, you can use the enter option which will prompt for new values.
    - e 076a:0000
10. To execute the program, you can use go option
    - G
11. After successful execution and termination of the program, you can check the result by checking the data memory segment
    - d 076a:0000
12. The result can be viewed in the respective address mentioned in the program.

3 a) Moving a string of bytes

Algorithm:

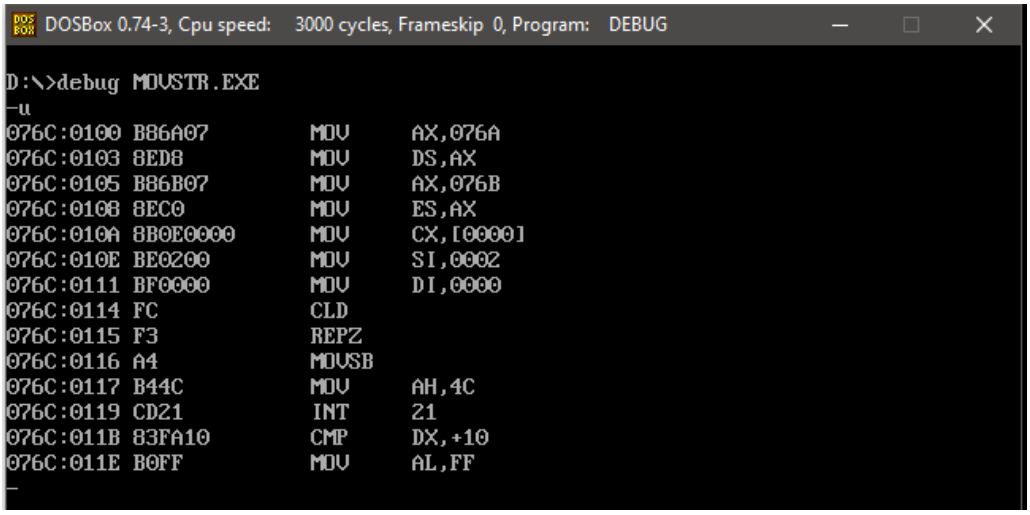
a) Assign extra to ax register
b) Load contents of memory location ax in register es
c) Load contents of memory location count in register cx
d) Load contents of memory location offset source in register si
e) Load contents of memory location offset dest in register di
f) Clear directional flag
g) move string from source to destination byte by byte using repeat
h) Load content 4ch termination code to ah register
i) Stops execution of the program

Program:

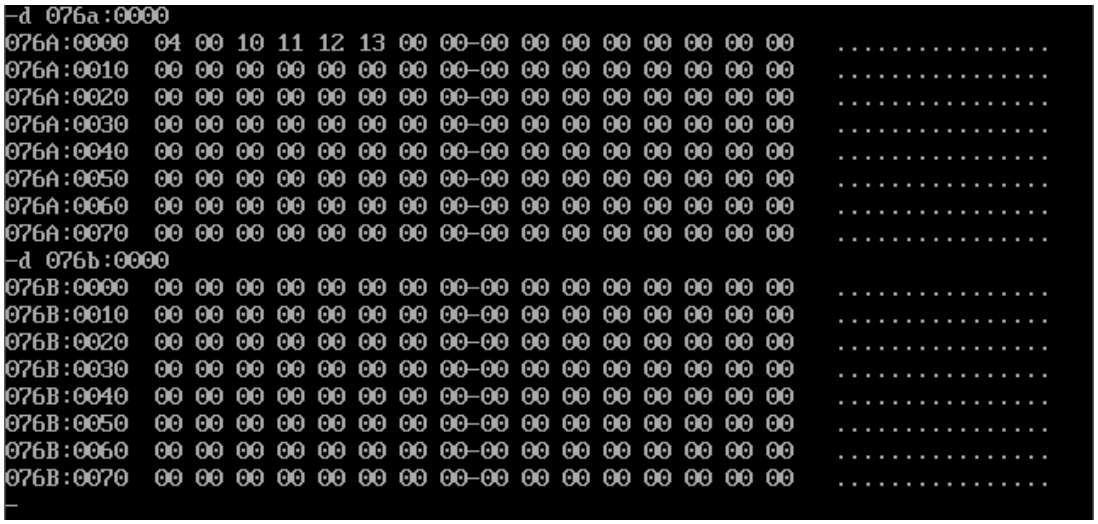
Program	**Comments**
assume cs:code,ds:data,es:extra	Initializing the code, data and extra segments to assembler
data segment	Data segment
count dw 0004h	count is declared and initialized to 0004h
source db 10h,11h,12h,13h	source is declared and initialized to 10h,11h,12h,13h
data ends	
extra segment	Extra segment
dest db 00h,00h,00h,00h	dest is declared and initialised to 00h,00h,00h,00h
extra 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 da
mov ax,extra	Load the data extra in to memory location of ax
mov es,ax	Transferring the data from memory location ax to es
mov cx,count	Transferring the data from memory location count to cx

mov si, offset source	Transferring the data from offset source to si
mov di, offset dest	Transferring the data from offset dest to di
cld	clear direction flag (LTR)
rep movsb	move string from source to destination byte by byte using repeat
mov ah,4ch	Transferring the termination code 4ch to ah
int 21h	Termination
code ends	Code ends
end start	

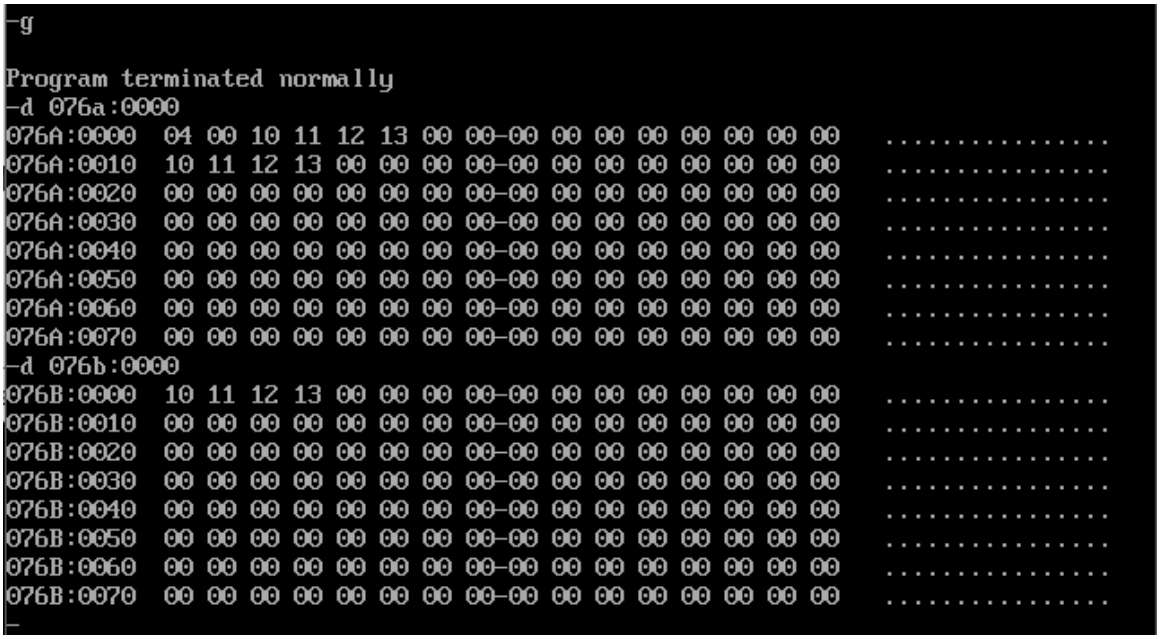
Unassembled code:



Sample Input :



Sample Input :



Result:

Thus, the assembly program for Moving a string of bytes is written and executed.

3 b) Comparing two string of bytes:

Algorithm:

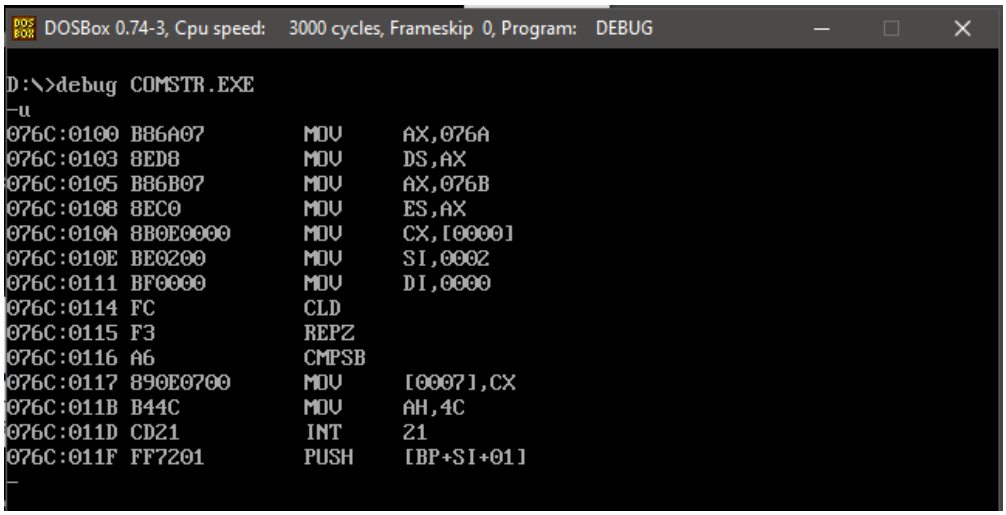
a) Assign extra to ax register
b) Load contents of memory location ax in register es
c) Load contents of memory location count in register cx
d) Load contents of memory location offset str1 in register si
e) Load contents of memory location offset str2 in register di
f) Clear directional flag
g) move string from source to destination byte by byte using repeat and compare
h) Load contents of memory location offset cx in status
i) Load content 4ch termination code to ah register
j) Stops execution of the program

Program:

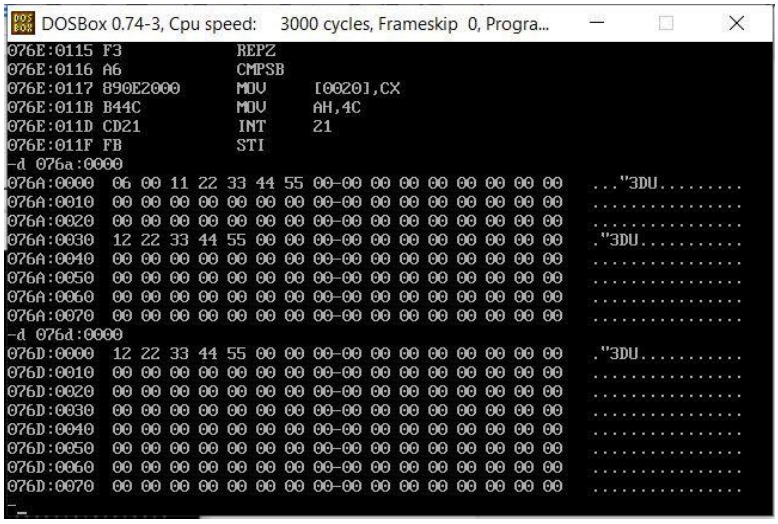
Program	**Comments**
assume cs:code,ds:data,es:extra	Initializing the code, data and extra segments to assembler
data segment	Data segment
count dw 0006h	count is declared and initialized to 0006h
str1 db 11h,22h,33h,44h,55h	Str1 is declared and initialized to 11h,22h,33h,44h,55h
status dw 0000h	status is declared and initialized to 0000h
data ends	
extra segment	Extra segment
str2 db 11h,22h,33h,44h,55h	Str2 is declared and initialised to 11h,22h,33h,44h,55h
extra 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 da
mov ax,extra	Load the data extra in to memory location of ax
mov es,ax	Transferring the data from memory location ax to es

mov cx,count	Transferring the data from memory location count to cx
mov si, offset str1	Transferring the data from offset str1 to si
mov di, offset str2	Transferring the data from offset str2 to di
cld	clear direction flag (LTR)
rep cmpsb	Compare string from source to destination byte by byte using repeat
mov status, cx	Transferring data from cx to status
mov ah,4ch	Transferring the termination code 4ch to ah
int 21h	Termination
code ends	Code ends
end start	

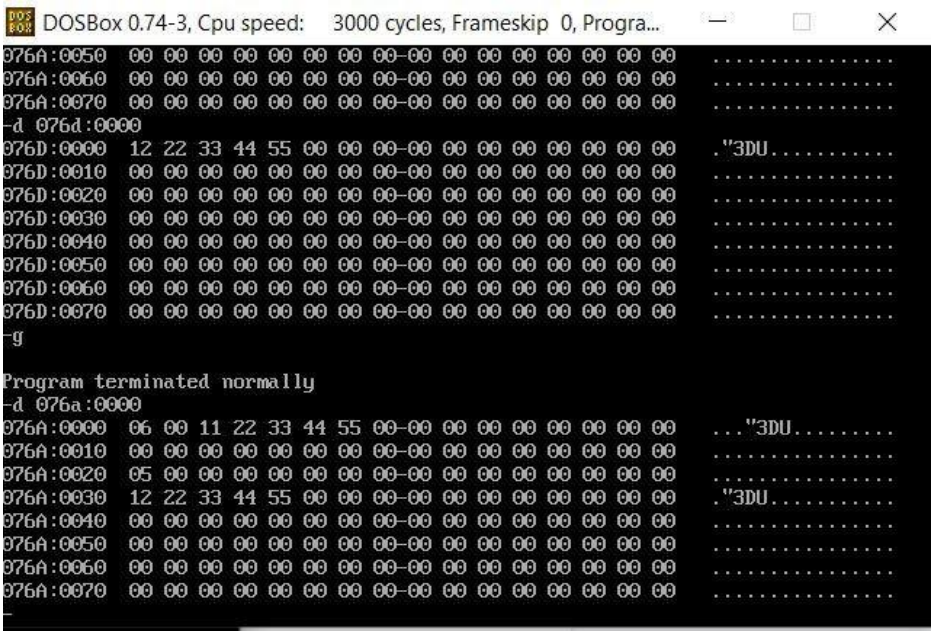
Unassembled code:



Sample Input :



Sample Input :



Result:

Thus, the assembly program for Comparing two string of bytes is written and executed.

3 c) Searching a byte in a string:

Algorithm:

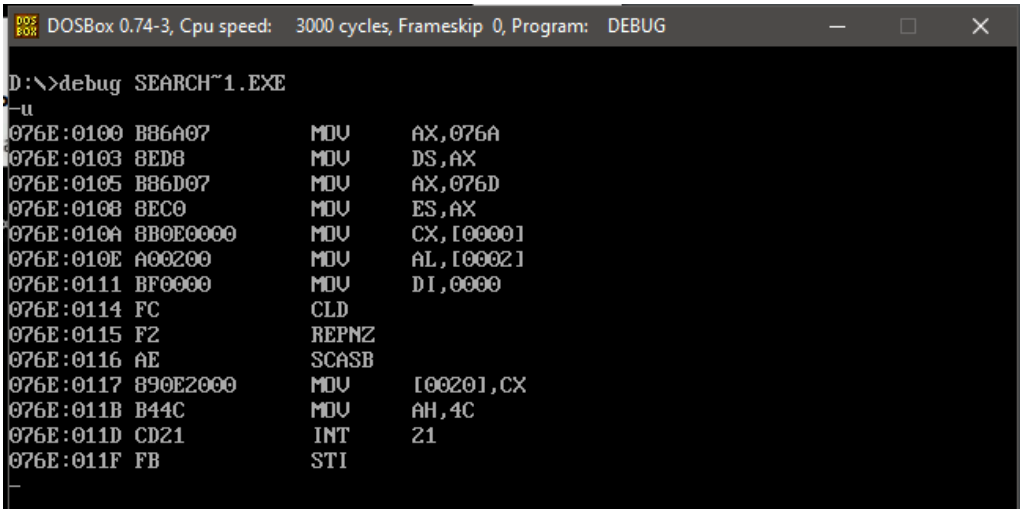
a) Assign extra to ax register
b) Load contents of memory location ax in register es
c) Load contents of memory location count in register cx
d) Load contents of memory location offset source in register si
e) Load contents of memory location offset dest in register di
f) Clear directional flag
g) move string from source to destination byte by byte using repeat
h) Load content 4ch termination code to ah register
i) Stops execution of the program

Program:

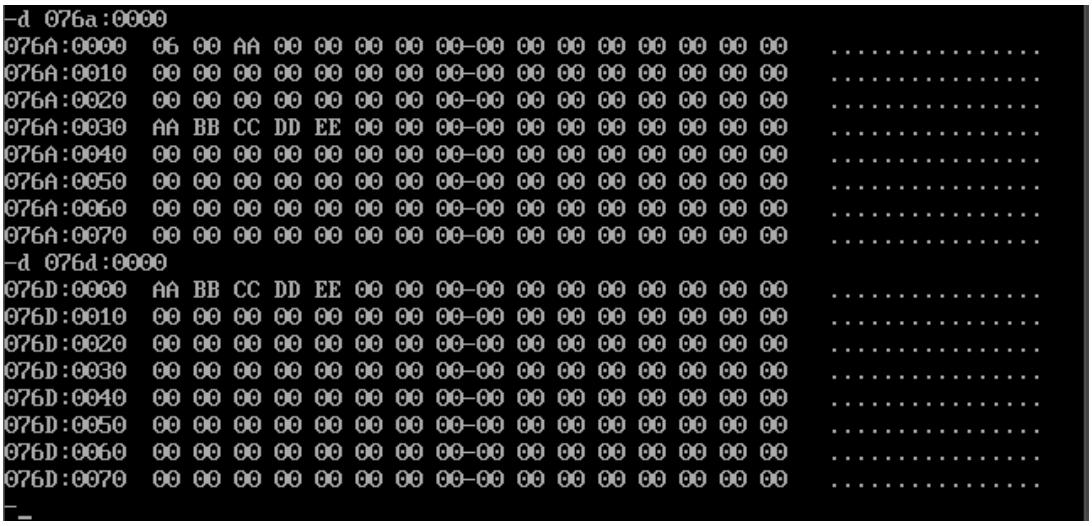
Program	**Comments**
assume cs:code,ds:data,es:extra	Initializing the code, data and extra segments to assembler
data segment	Data segment
count dw 0006h	count is declared and initialized to 0006h
str1 db 0aah	Str1 is declared and initialized to 0aah
org 0020h	org is declared and initialized to 0020h
status dw 0000h	status is declared and initialized to 0000h
data ends	
extra segment	Extra segment
str2 db 0aah,0bbh,0cch,0ddh,0eeh	Str2 is declared and initialised to 0aah,0bbh,0cch,0ddh,0eeh
extra 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 da
mov ax,extra	Load the data extra in to memory location of ax

mov es,ax	Transferring the data from memory location ax to es
mov cx,count	Transferring the data from memory location count to cx
mov al,str1	Transferring the data from offset str1 to al
mov di,offset str2	Transferring the data from offset str2 to di
cld	clear direction flag (LTR)
rep scasb	Searching the source string byte by byte using repne
mov status, cx	Transferring data from cx to status
mov ah,4ch	Transferring the termination code 4ch to ah
int 21h	Termination
code ends	Code ends
end start	

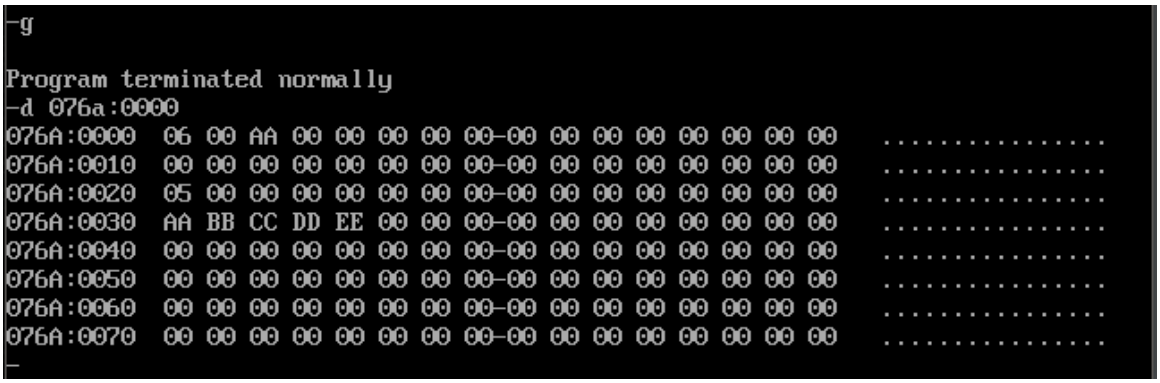
Unassembled code:



Sample Input :



Sample Input :



Result:

Thus, the assembly program for Searching a byte in a string is written and executed.

3 d) Moving a string without using string operations:

Algorithm:

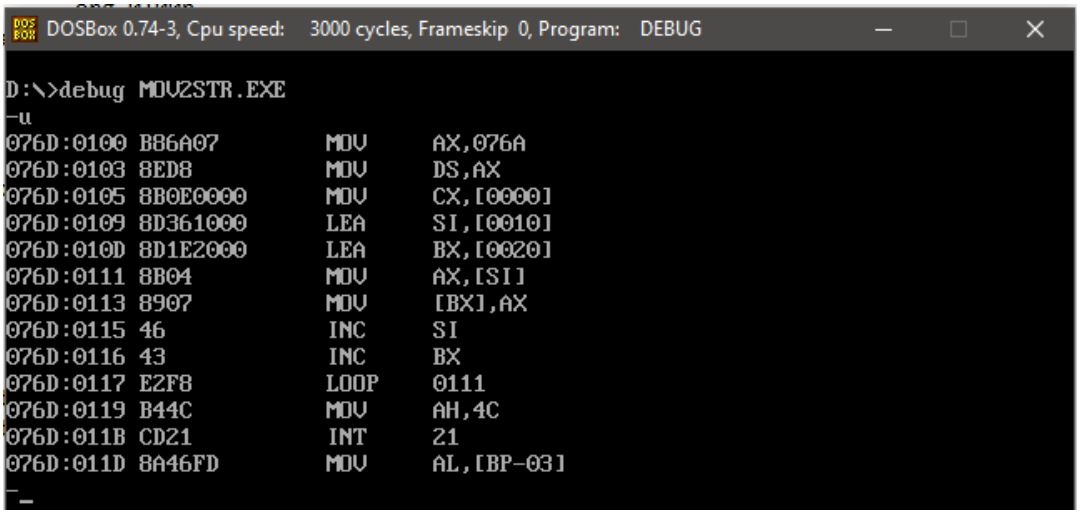
a) Load contents of memory location count in register cx
b) Load eff_address of str1 to si .
c) Load eff_address of str2 to bx .
d) Moving value of [si] to ax
e) Moving value of ax to [bx]
f) Si++ and bx++
g) Loop_a
h) Load content 4ch termination code to ah register
i) Stops execution of the program

Program:

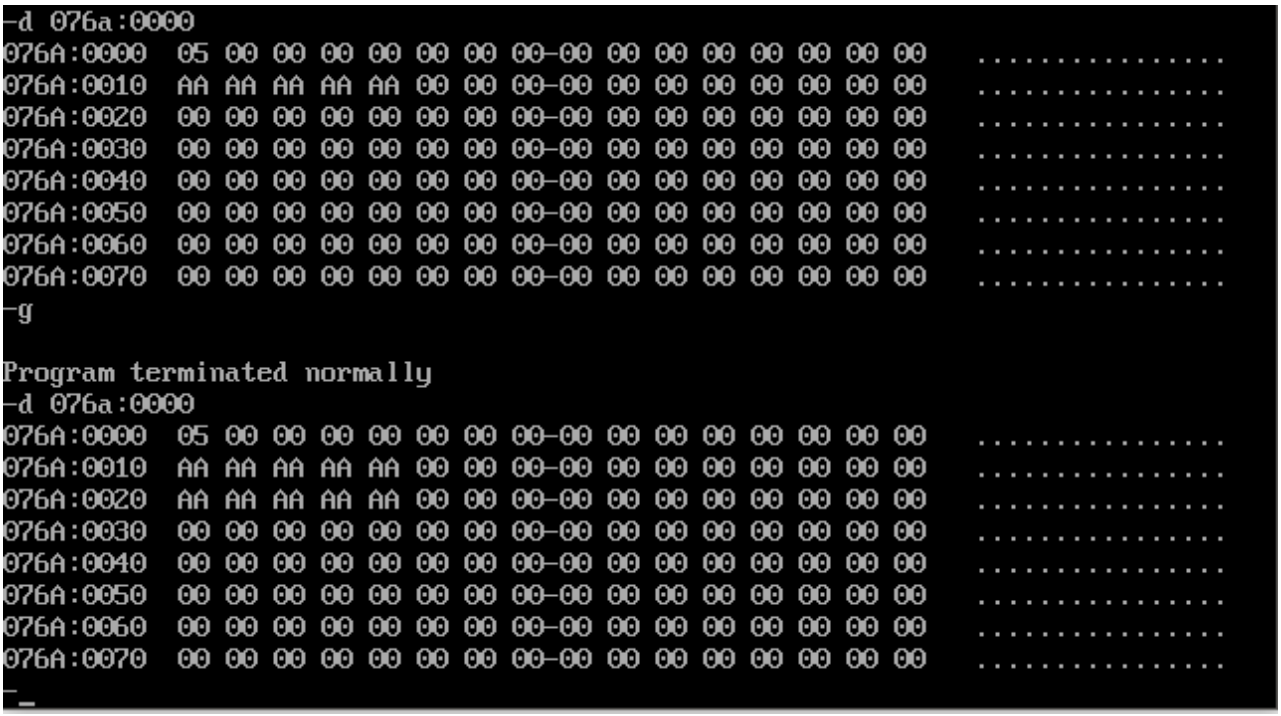
Program	**Comments**
assume cs:code,ds:data,es:extra	Initializing the code, data and extra segments to assembler
data segment	Data segment
count dw 0005h	count is declared and initialized to 0005h
org 0010h	org is declared and initialized to 0010h
str1 db 011h,022h,033h,044h,055h	Str1 is declared and initialized to 011h,022h,033h,044h,055h
org 0020h	org is declared and initialized to 0020h
Str2 db 00h,00h,00h,00h,00h	source is declared and initialized to 00h,00h,00h,00h,00h
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 da
mov cx,count	Load the data count in to memory location of cx
lea si,str1	Load eff_address of str1 to si .
lea bx,str2	Load eff_address of str2 to bx .

loop_a: mov ax,[si]	Moving value of [si] to ax
mov [bx],ax	Moving value of ax to [bx]
inc si	Si++
inc bx	Bx++
loop loop_a	loop again to loop_a
mov ah,4ch	Transferring the termination code 4ch to ah
int 21h	Termination
code ends	Code ends
end start	

Unassembled code:



Sample Input and output:



Result:

Thus, the assembly program for Moving a string without using string operations is written and executed.