String Manipulations

Expt No:	3	**Name:**	Shivanirudh S G
Date :	11/09/2020	**Reg No:**	185001146

Aim:

To perform string manipulation in 8086.

Moving a string of bytes

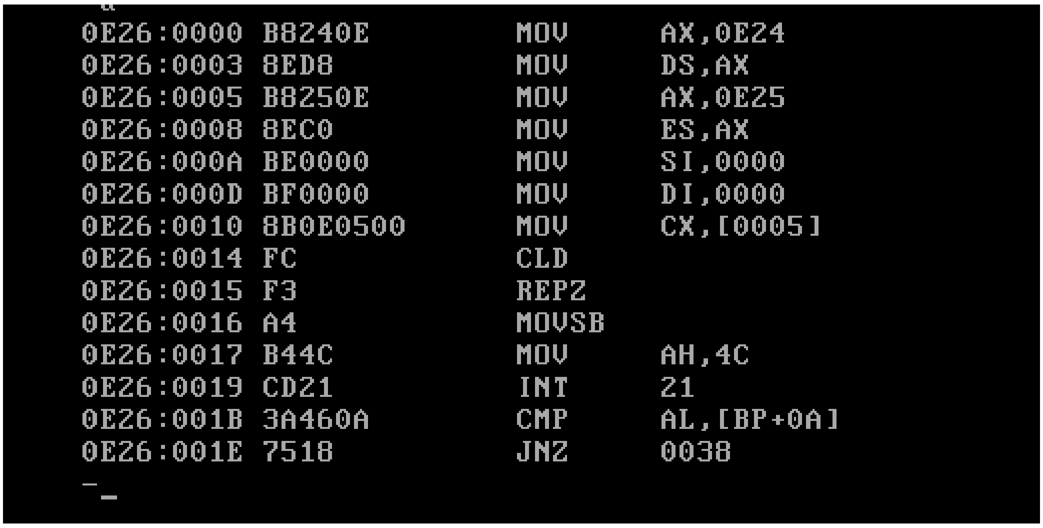
Algorithm:

- Move the data segment to the AX register and then move it to the DS register.
- Move the extra segment to the AX register and then move it to the ES register.
- Move the offset of source string to SI register.
- Move the offset of destination string to the DI register.
- Move the length of the string to the CX register.
- Clear the direction flag.
- Move the string from address pointed by DS and SI to address pointed by ES and DI using REP MOVSB.

Program:

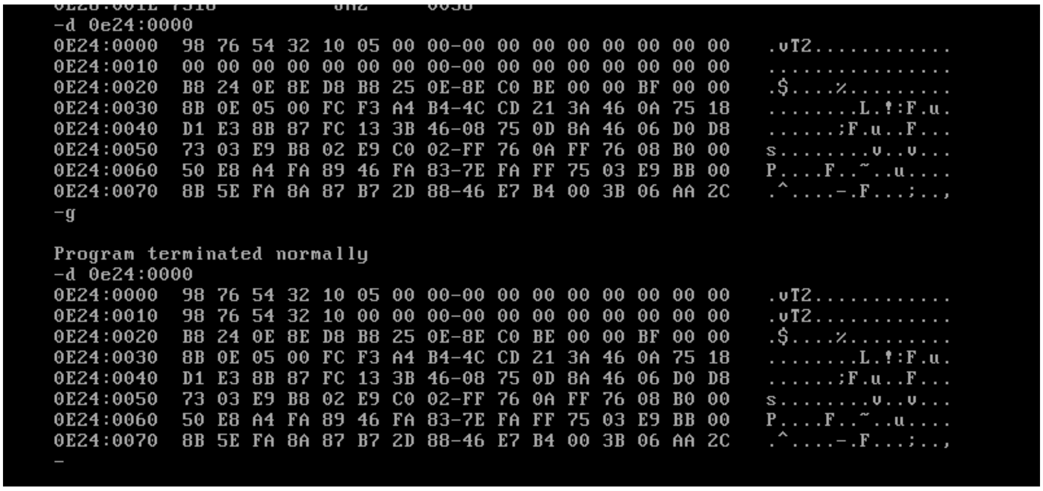
Program	Comments
assume cs:code, ds:data, es:extra	Declare code, data and extra segments
data segment	Start of data segment
source db 98H, 76H, 54H, 32H, 10H	Define array of bytes source
count dw 0005H	Define word count with hex value 0005
data ends	End of data segment
extra segment	Start of extra segment
dest db ?	Define array of bytes dest with unknown values
extra ends	End of extra segment
code segment	Start of code segment
start: mov ax, data	Move data segment contents to AX register
mov ds, ax	Move data in AX register to DS register
mov ax, extra	Move extra segment contents to AX register
mov es, ax	Move data in AX register to ES register
mov si, offset source	Move offset of source to SI register
mov di, offset dest	Move offset of dest to DI register
mov cx, count	Move value of count to CX register
cld	Clear the direction flag
rep movsb	Move contents of SI to DI till CX = 0
int 21h	Request interrupt routine
code ends	End of code segment
end start	

Unassembled code:



```
0E26:0000 B8240E        MOV     AX,0E24
0E26:0003 8ED8          MOV     DS,AX
0E26:0005 B8250E        MOV     AX,0E25
0E26:0008 8EC0          MOV     ES,AX
0E26:000A BE0000        MOV     SI,0000
0E26:000D BF0000        MOV     DI,0000
0E26:0010 8B0E0500      MOV     CX,[0005]
0E26:0014 FC            CLD
0E26:0015 F3            REPZ
0E26:0016 A4            MOVSB
0E26:0017 B44C          MOV     AH,4C
0E26:0019 CD21          INT     21
0E26:001B 3A460A        CMP     AL,[BP+0A]
0E26:001E 7518          JNZ     0038
-_
```

Input and Output:



```
-d 0e24:0000
0E24:0000  98 76 54 32 10 05 00 00-00 00 00 00 00 00 00 00   .vT2............
0E24:0010  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
0E24:0020  B8 24 0E 8E D8 B8 25 0E-8E C0 BE 00 00 BF 00 00   .$....%.........
0E24:0030  8B 0E 05 00 FC F3 A4 B4-4C CD 21 3A 46 0A 75 18   ........L.!:F.u.
0E24:0040  D1 E3 8B 87 FC 13 3B 46-08 75 0D 8A 46 06 D0 D8   ......;F.u..F...
0E24:0050  73 03 E9 B8 02 E9 C0 02-FF 76 0A FF 76 08 B0 00   s........v..v...
0E24:0060  50 E8 A4 FA 89 46 FA 83-7E FA FF 75 03 E9 BB 00   P....F..~..u....
0E24:0070  8B 5E FA 8A 87 B7 2D 88-46 E7 B4 00 3B 06 AA 2C   .^....-.F...;..,
-g

Program terminated normally
-d 0e24:0000
0E24:0000  98 76 54 32 10 05 00 00-00 00 00 00 00 00 00 00   .vT2............
0E24:0010  98 76 54 32 10 00 00 00-00 00 00 00 00 00 00 00   .vT2............
0E24:0020  B8 24 0E 8E D8 B8 25 0E-8E C0 BE 00 00 BF 00 00   .$....%.........
0E24:0030  8B 0E 05 00 FC F3 A4 B4-4C CD 21 3A 46 0A 75 18   ........L.!:F.u.
0E24:0040  D1 E3 8B 87 FC 13 3B 46-08 75 0D 8A 46 06 D0 D8   ......;F.u..F...
0E24:0050  73 03 E9 B8 02 E9 C0 02-FF 76 0A FF 76 08 B0 00   s........v..v...
0E24:0060  50 E8 A4 FA 89 46 FA 83-7E FA FF 75 03 E9 BB 00   P....F..~..u....
0E24:0070  8B 5E FA 8A 87 B7 2D 88-46 E7 B4 00 3B 06 AA 2C   .^....-.F...;..,
-
```

Figure 1: **Input:** *DS:* 98 76 54 32 10H, *ES:* 00 00 00 00 00H;
Output: *DS:* 98 76 54 32 10H, *ES:* 98 76 54 32 10H

Comparing 2 strings of bytes

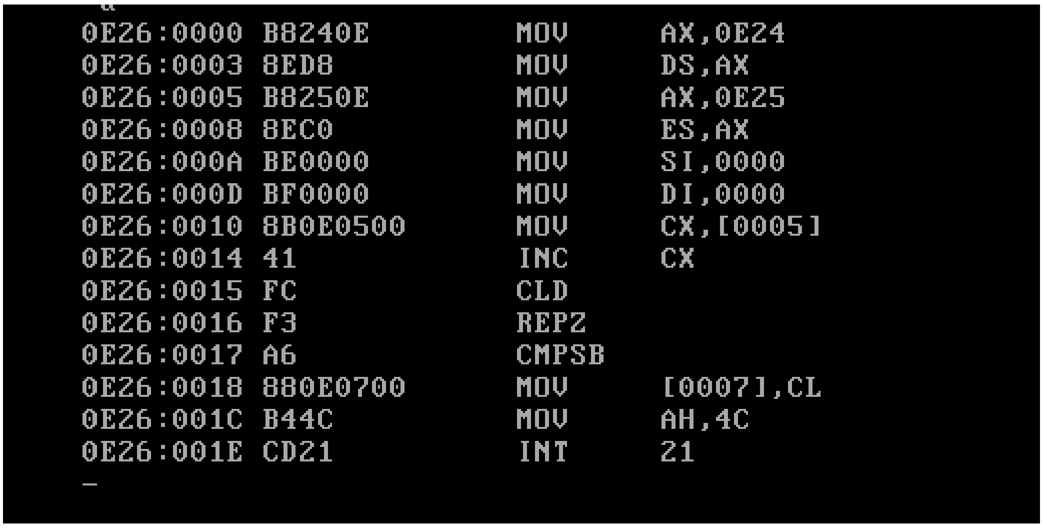
Algorithm:

- Move the data segment to the AX register and then move it to the DS register.
- Move the extra segment to the AX register and then move it to the ES register.
- Move the offset of string s1 to SI register.
- Move the offset of string s2 to the DI register.
- Move the length of the string to the CX register. Increment value of CX.
- Clear the direction flag.
- Compare the strings from addresses pointed by DS and SI and by ES and DI using REPE CMPSB.
- Move contents of cx, which now has the position in string where the difference is, to RESULT.

Program:

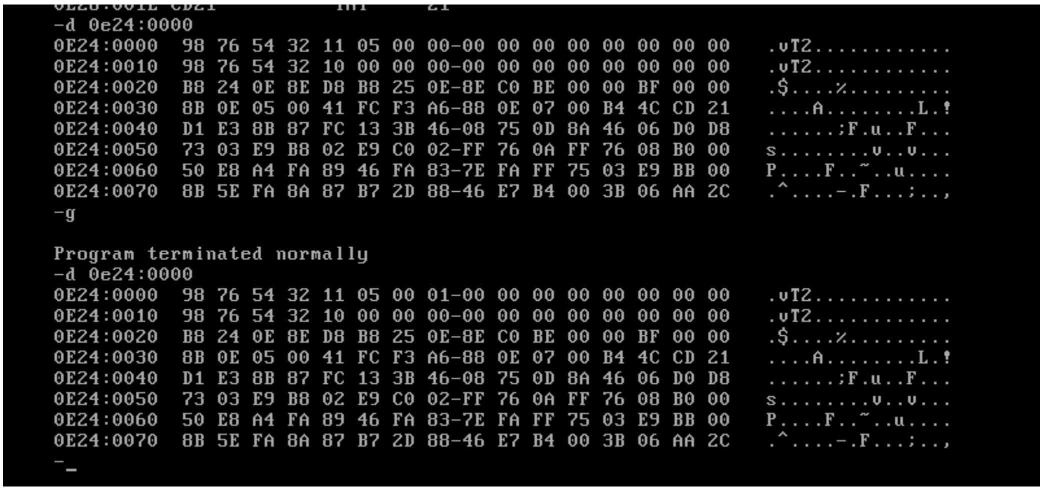
Program	Comments
assume cs:code, ds:data, es:extra	Declare code, data and extra segments
data segment	Start of data segment
s1 db 98H, 76H, 54H, 32H, 11H	Define array of bytes s1
count dw 0005H	Define word count with hex value 0005
result db 00H	Define byte result with hex value 00
data ends	End of data segment
extra segment	Start of extra segment
s2 db 98H, 76H, 54H, 32H, 10H	Define array of bytes s2
extra ends	End of extra segment
code segment	Start of code segment
start: mov ax, data	Move data segment contents to AX register
mov ds, ax	Move data in AX register to DS register
mov ax, extra	Move extra segment contents to AX register
mov es, ax	Move data in AX register to ES register
mov si, offset source	Move offset of source to SI register
mov di, offset dest	Move offset of dest to DI register
mov cx, count	Move value of count to CX register
inc cx	Increment value of CX
cld	Clear the direction flag
repe cmpsb	Compare bytes of s1 and s2 till unequal
mov result, cx	Move contents of CX to result
int 21h	Request interrupt routine
code ends	End of code segment
end start	

Unassembled code:



```
 u
0E26:0000 B8240E        MOV     AX,0E24
0E26:0003 8ED8          MOV     DS,AX
0E26:0005 B8250E        MOV     AX,0E25
0E26:0008 8EC0          MOV     ES,AX
0E26:000A BE0000        MOV     SI,0000
0E26:000D BF0000        MOV     DI,0000
0E26:0010 8B0E0500      MOV     CX,[0005]
0E26:0014 41            INC     CX
0E26:0015 FC            CLD
0E26:0016 F3            REPZ
0E26:0017 A6            CMPSB
0E26:0018 880E0700      MOV     [0007],CL
0E26:001C B44C          MOV     AH,4C
0E26:001E CD21          INT     21
-
```

Input and Output:



```
0E26:001E CD21          INT     21
-d 0e24:0000
0E24:0000  98 76 54 32 11 05 00 00-00 00 00 00 00 00 00 00   .vTZ............
0E24:0010  98 76 54 32 10 00 00 00-00 00 00 00 00 00 00 00   .vTZ............
0E24:0020  B8 24 0E 8E D8 B8 25 0E-8E C0 BE 00 00 BF 00 00   .$....%.........
0E24:0030  8B 0E 05 00 41 FC F3 A6-88 0E 07 00 B4 4C CD 21   ....A........L.!
0E24:0040  D1 E3 8B 87 FC 13 3B 46-08 75 0D 8A 46 06 D0 D8   ......;F.u..F...
0E24:0050  73 03 E9 B8 02 E9 C0 02-FF 76 0A FF 76 08 B0 00   s........v..v...
0E24:0060  50 E8 A4 FA 89 46 FA 83-7E FA FF 75 03 E9 BB 00   P....F..~..u....
0E24:0070  8B 5E FA 8A 87 B7 2D 88-46 E7 B4 00 3B 06 AA 2C   .^....-.F...;..,
-g

Program terminated normally
-d 0e24:0000
0E24:0000  98 76 54 32 11 05 00 01-00 00 00 00 00 00 00 00   .vTZ............
0E24:0010  98 76 54 32 10 00 00 00-00 00 00 00 00 00 00 00   .vTZ............
0E24:0020  B8 24 0E 8E D8 B8 25 0E-8E C0 BE 00 00 BF 00 00   .$....%.........
0E24:0030  8B 0E 05 00 41 FC F3 A6-88 0E 07 00 B4 4C CD 21   ....A........L.!
0E24:0040  D1 E3 8B 87 FC 13 3B 46-08 75 0D 8A 46 06 D0 D8   ......;F.u..F...
0E24:0050  73 03 E9 B8 02 E9 C0 02-FF 76 0A FF 76 08 B0 00   s........v..v...
0E24:0060  50 E8 A4 FA 89 46 FA 83-7E FA FF 75 03 E9 BB 00   P....F..~..u....
0E24:0070  8B 5E FA 8A 87 B7 2D 88-46 E7 B4 00 3B 06 AA 2C   .^....-.F...;..,
-_
```

Figure 2: **Input:** *DS:* 98 76 54 32 11H, *ES:* 98 76 54 32 10H; **Output:** *Result:* 0001

Searching a byte in a string

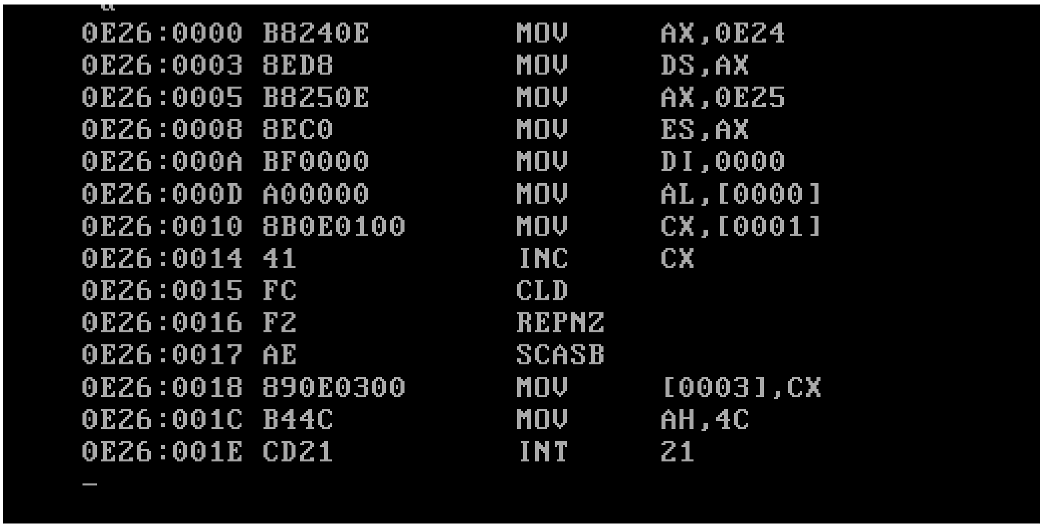
Algorithm:

- Move the data segment to the AX register and then move it to the DS register.
- Move the extra segment to the AX register and then move it to the ES register.
- Move the offset of string to the DI register.
- Move the byte to be searched to the AL register.
- Move the length of the string to the CX register. Increment value of CX
- Clear the direction flag.
- Scan the string for the specified byte using REPNE SCASB.
- Move contents of CX register, which now has location of the specified byte in the string, to RESULT.

Program:

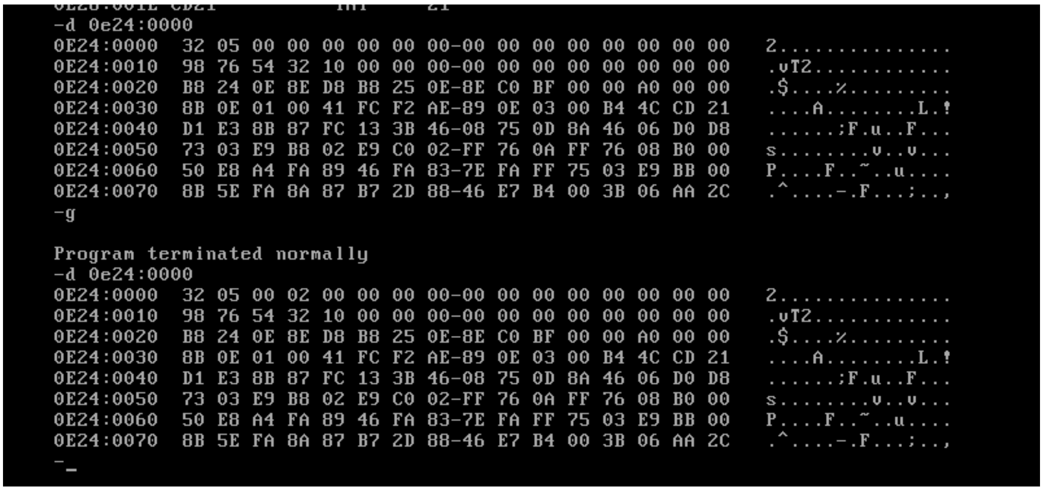
Program	**Comments**
assume cs:code, ds:data, es:extra	Declare code, data and extra segments
data segment	Start of data segment
val db 32H	Define byte val with hex value 32
count dw 0005H	Define word count with hex value 0005
result dw 0000H	Define word result with hex value 0000
data ends	End of data segment
extra segment	Start of extra segment
str db 98H, 76H, 54H, 32H, 10H	Define array of bytes dest with unknown values
extra ends	End of extra segment
code segment	Start of code segment
start: mov ax, data	Move data segment contents to AX register
mov ds, ax	Move data in AX register to DS register
mov ax, extra	Move extra segment contents to AX register
mov es, ax	Move data in AX register to ES register
mov di, offset str	Move offset of str to DI register
mov al, val	Move value of val to AL register
mov cx, count	Move value of count to CX register
inc cx	Increment value of CX
cld	Clear the direction flag
repne scasb	Scan contents of DI for AL till CX = 0
mov result, cx	Move contents of CX to result
int 21h	Request interrupt routine
code ends	End of code segment
end start	

Unassembled code:



```
 -u
   0E26:0000 B8240E         MOV     AX,0E24
   0E26:0003 8ED8           MOV     DS,AX
   0E26:0005 B8250E         MOV     AX,0E25
   0E26:0008 8EC0           MOV     ES,AX
   0E26:000A BF0000         MOV     DI,0000
   0E26:000D A00000         MOV     AL,[0000]
   0E26:0010 8B0E0100       MOV     CX,[0001]
   0E26:0014 41             INC     CX
   0E26:0015 FC             CLD
   0E26:0016 F2             REPNZ
   0E26:0017 AE             SCASB
   0E26:0018 890E0300       MOV     [0003],CX
   0E26:001C B44C           MOV     AH,4C
   0E26:001E CD21           INT     21
   -
```

Input and Output:



```
0E26:001E CD21           INT     21
-d 0e24:0000
0E24:0000  32 05 00 00 00 00 00 00-00 00 00 00 00 00 00 00   2...............
0E24:0010  98 76 54 32 10 00 00 00-00 00 00 00 00 00 00 00   .vT2............
0E24:0020  B8 24 0E 8E D8 B8 25 0E-8E C0 BF 00 00 A0 00 00   .$....%.........
0E24:0030  8B 0E 01 00 41 FC F2 AE-89 0E 03 00 B4 4C CD 21   ....A.......L.!
0E24:0040  D1 E3 8B 87 FC 13 3B 46-08 75 0D 8A 46 06 D0 D8   ......;F.u..F...
0E24:0050  73 03 E9 B8 02 E9 C0 02-FF 76 0A FF 76 08 B0 00   s........v..v...
0E24:0060  50 E8 A4 FA 89 46 FA 83-7E FA FF 75 03 E9 BB 00   P....F..~..u....
0E24:0070  8B 5E FA 8A 87 B7 2D 88-46 E7 B4 00 3B 06 AA 2C   .^....-.F...;..,
-g

Program terminated normally
-d 0e24:0000
0E24:0000  32 05 00 02 00 00 00 00-00 00 00 00 00 00 00 00   2...............
0E24:0010  98 76 54 32 10 00 00 00-00 00 00 00 00 00 00 00   .vT2............
0E24:0020  B8 24 0E 8E D8 B8 25 0E-8E C0 BF 00 00 A0 00 00   .$....%.........
0E24:0030  8B 0E 01 00 41 FC F2 AE-89 0E 03 00 B4 4C CD 21   ....A.......L.!
0E24:0040  D1 E3 8B 87 FC 13 3B 46-08 75 0D 8A 46 06 D0 D8   ......;F.u..F...
0E24:0050  73 03 E9 B8 02 E9 C0 02-FF 76 0A FF 76 08 B0 00   s........v..v...
0E24:0060  50 E8 A4 FA 89 46 FA 83-7E FA FF 75 03 E9 BB 00   P....F..~..u....
0E24:0070  8B 5E FA 8A 87 B7 2D 88-46 E7 B4 00 3B 06 AA 2C   .^....-.F...;..,
-_
```

Figure 3: **Input:** *DS:* Value: 32H, *ES:* 98 76 54 32 10H;
Output: *Result:* 0002H

Moving a string without using string instructions

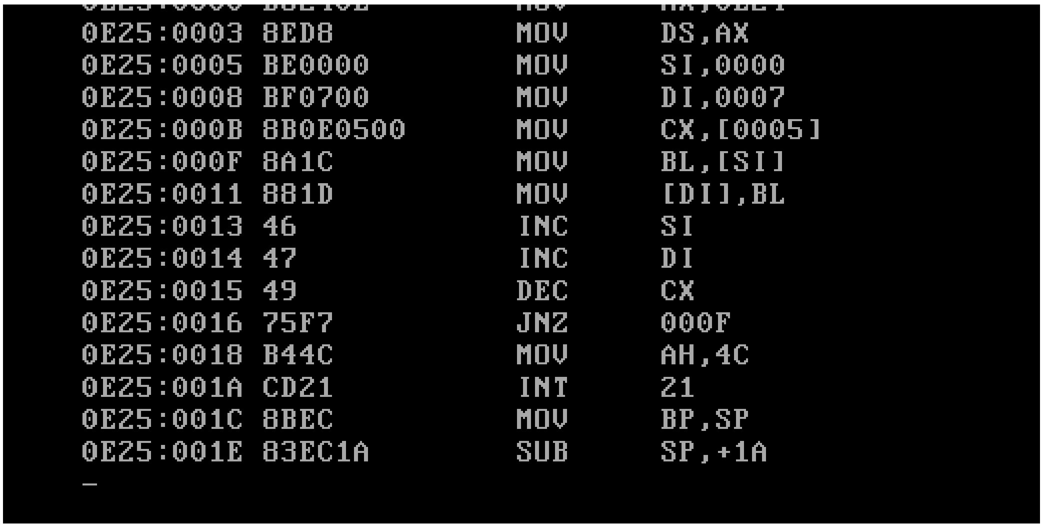
Algorithm:

- Move the data segment to the AX register and then move it to the DS register.
- Move the offset of source string to the SI register.
- Move the offset of destination string to the DI register.
- Move the length of the string to the CX register.
- Repeat till CX is 0:
  - Move content of SI to BL, and then to DI.
  - Increment SI, DI.
  - Decrement CX.

Program:

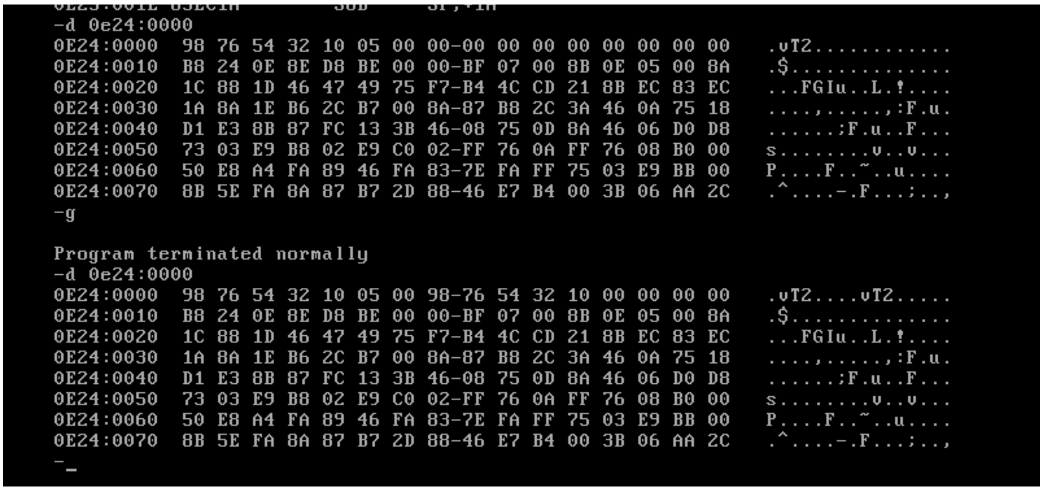
Program	Comments
assume cs:code, ds:data	Declare code and data segments
data segment	Start of data segment
source db 98H, 76H, 54H, 32H, 10H	Define array of bytes source
count dw 0005H	Define word count with hex value 0005
dest db ?	Define array of bytes dest with unknown values
data ends	End of data segment
code segment	Start of code segment
start: mov ax, data	Move data segment contents to AX register
mov ds, ax	Move data in AX register to DS register
mov si, offset source	Move offset of source to SI register
mov di, offset dest	Move offset of dest to DI register
mov cx, count	Move value of count to CX register
here: mov bl, [si]	Move contents of SI to BL register
mov [di], bl	Move contents of BL register to DI
inc si	Increment value of SI
inc di	Increment value of DI
dec cx	Decrement value of CX
int 21h	Request interrupt routine
code ends	End of code segment
end start	

Unassembled code:



```
0E25:0003 8ED8          MOV     DS,AX
0E25:0005 BE0000        MOV     SI,0000
0E25:0008 BF0700        MOV     DI,0007
0E25:000B 8B0E0500      MOV     CX,[0005]
0E25:000F 8A1C          MOV     BL,[SI]
0E25:0011 881D          MOV     [DI],BL
0E25:0013 46            INC     SI
0E25:0014 47            INC     DI
0E25:0015 49            DEC     CX
0E25:0016 75F7          JNZ     000F
0E25:0018 B44C          MOV     AH,4C
0E25:001A CD21          INT     21
0E25:001C 8BEC          MOV     BP,SP
0E25:001E 83EC1A        SUB     SP,+1A
-
```

Input and Output:



```
-d 0e24:0000
0E24:0000  98 76 54 32 10 05 00 00-00 00 00 00 00 00 00 00   .vTZ............
0E24:0010  B8 24 0E 8E D8 BE 00 00-BF 07 00 8B 0E 05 00 8A   .$..............
0E24:0020  1C 88 1D 46 47 49 75 F7-B4 4C CD 21 8B EC 83 EC   ...FGIu..L.!....
0E24:0030  1A 8A 1E B6 2C B7 00 8A-87 B8 2C 3A 46 0A 75 18   ....,.....,:F.u.
0E24:0040  D1 E3 8B 87 FC 13 3B 46-08 75 0D 8A 46 06 D0 D8   ......;F.u..F...
0E24:0050  73 03 E9 B8 02 E9 C0 02-FF 76 0A FF 76 08 B0 00   s........v..v...
0E24:0060  50 E8 A4 FA 89 46 FA 83-7E FA FF 75 03 E9 BB 00   P....F..~..u....
0E24:0070  8B 5E FA 8A 87 B7 2D 88-46 E7 B4 00 3B 06 AA 2C   .^....-.F...;..,
-g

Program terminated normally
-d 0e24:0000
0E24:0000  98 76 54 32 10 05 00 98-76 54 32 10 00 00 00 00   .vTZ....vTZ.....
0E24:0010  B8 24 0E 8E D8 BE 00 00-BF 07 00 8B 0E 05 00 8A   .$..............
0E24:0020  1C 88 1D 46 47 49 75 F7-B4 4C CD 21 8B EC 83 EC   ...FGIu..L.!....
0E24:0030  1A 8A 1E B6 2C B7 00 8A-87 B8 2C 3A 46 0A 75 18   ....,.....,:F.u.
0E24:0040  D1 E3 8B 87 FC 13 3B 46-08 75 0D 8A 46 06 D0 D8   ......;F.u..F...
0E24:0050  73 03 E9 B8 02 E9 C0 02-FF 76 0A FF 76 08 B0 00   s........v..v...
0E24:0060  50 E8 A4 FA 89 46 FA 83-7E FA FF 75 03 E9 BB 00   P....F..~..u....
0E24:0070  8B 5E FA 8A 87 B7 2D 88-46 E7 B4 00 3B 06 AA 2C   .^....-.F...;..,
-_
```

Figure 4: **Input:** *source:* 98 76 54 32 10H, *dest:* 00 00 00 00 00H; **Output:** *source:* 98 76 54 32 10H, *dest:* 98 76 54 32 10H