

Name:	Pankhania Aanandi R.
Roll No:	IT081
Batch	I1

### Experiment 3

#### **AIM: Study of string related instructions**

1. **Write an assembly language program for moving a string from one segment to another segment.**

**Rules for Operands:** You have to use your name as a string name.

e.g. myname DB "Sunil K. Vithlani\$"

**Write your code here:**

DATA SEGMENT

myname db 'AANANDI R. PANKHANIAS'

len equ \$-myname

DATA ENDS

DATA1 SEGMENT

str1 db 15 DUP(0)

DATA1 ENDS

CODE SEGMENT

assume CS:CODE ,DS:DATA , ES:DATA1

Start : mov AX,DATA

mov DS,AX

mov AX,DATA1

mov ES, AX

LEA SI,myname

LEA DI,str1

mov CX,len

CLD

REP movsb

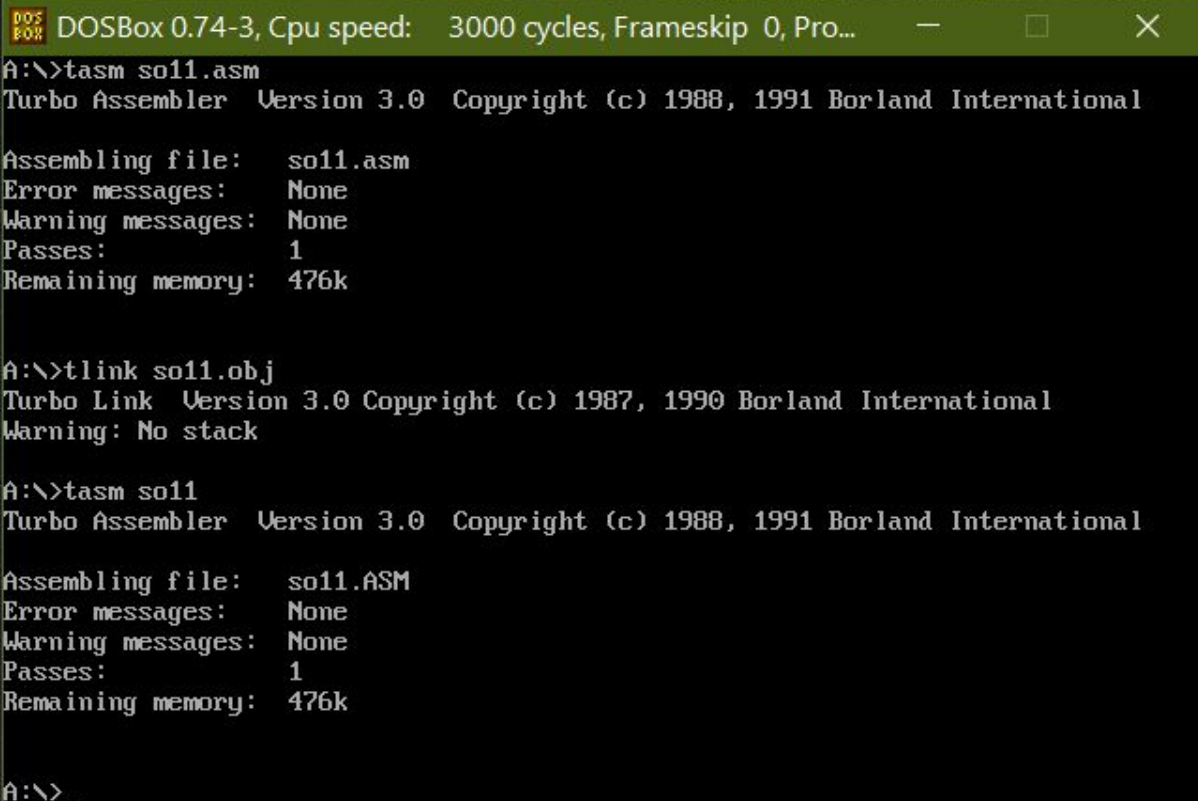
INT 3h

CODE ENDS

END Start

### Compilation /Running and Debugging steps:

(As given in the lab manual as an example of multiplication program on page no:5 of lab manual)



```

DOSBOX 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Pro...
A:\>tasm so11.asm
Turbo Assembler Version 3.0 Copyright (c) 1988, 1991 Borland International

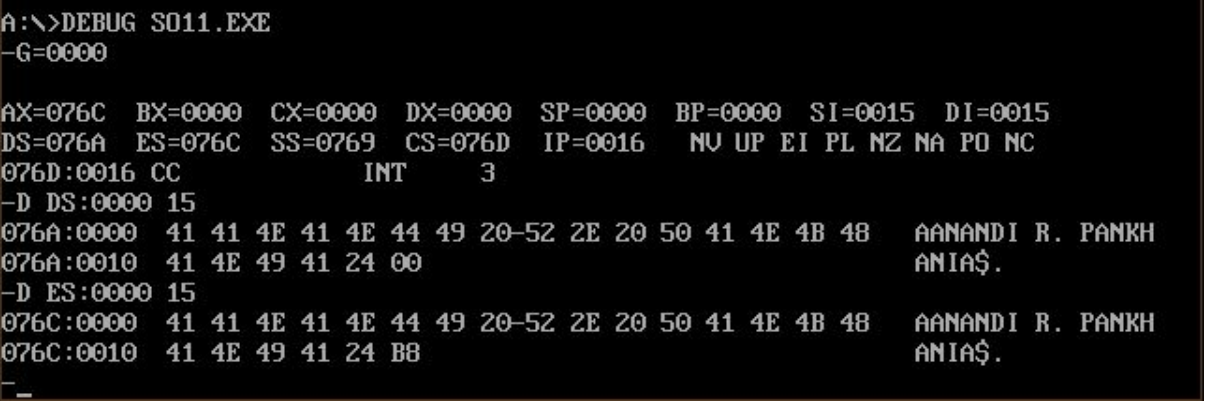
Assembling file:    so11.asm
Error messages:     None
Warning messages:   None
Passes:             1
Remaining memory:   476k

A:\>tlink so11.obj
Turbo Link Version 3.0 Copyright (c) 1987, 1990 Borland International
Warning: No stack

A:\>tasm so11
Turbo Assembler Version 3.0 Copyright (c) 1988, 1991 Borland International

Assembling file:    so11.ASM
Error messages:     None
Warning messages:   None
Passes:             1
Remaining memory:   476k

A:\>_
  
```



```

A:\>DEBUG SO11.EXE
-G=0000

AX=076C BX=0000 CX=0000 DX=0000 SP=0000 BP=0000 SI=0015 DI=0015
DS=076A ES=076C SS=0769 CS=076D IP=0016  NV UP EI PL NZ NA PO NC
076D:0016 CC          INT     3
-D DS:0000 15
076A:0000 41 41 4E 41 4E 44 49 20-52 2E 20 50 41 4E 4B 48  AANANDI R. PANKH
076A:0010 41 4E 49 41 24 00                                ANIA$.
-D ES:0000 15
076C:0000 41 41 4E 41 4E 44 49 20-52 2E 20 50 41 4E 4B 48  AANANDI R. PANKH
076C:0010 41 4E 49 41 24 B8                                ANIA$.
_
  
```

**Output:**

Screenshots of memory location before moving and after moving a string. (output of -d ds:offset\_address command.)

```

A:\>DEBUG S011.EXE
-G=0000

AX=076C BX=0000 CX=0000 DX=0000 SP=0000 BP=0000 SI=0015 DI=0015
DS=076A ES=076C SS=0769 CS=076D IP=0016 NU UP EI PL NZ NA PO NC
076D:0016 CC INT 3
-D DS:0000 15
076A:0000 41 41 4E 41 4E 44 49 20 52 2E 20 50 41 4E 4B 48 AANANDI R. PANKH
076A:0010 41 4E 49 41 24 00 ANIA$.
-D ES:0000 15
076C:0000 41 41 4E 41 4E 44 49 20 52 2E 20 50 41 4E 4B 48 AANANDI R. PANKH
076C:0010 41 4E 49 41 24 B8 ANIA$.

```

## 2. Write an assembly language program to compare two strings of equal length.

### Rules for Operands:

Case1: Take your name as both string and show results.

E.g str1 DB "sunil" and str2 DB "sunil"

Case2: Take your name as a upper case in 1<sup>st</sup> string and as a lower case in 2<sup>nd</sup> string.

E.g. str1 DB "SUNIL" and str2 DB "sunil"

### Write your code here:

DATA SEGMENT

DEMO DB 'aanandi\$'

STRNG DB 'aanandi\$'

msg1 DB 'strings are same\$'

msg2 DB 'strings are not same\$'

DATA ENDS

CODE SEGMENT

assume CS:CODE,DS:DATA,ES:DATA

start:mov AX,DATA

mov DS,AX

```
mov ES,AX
LEA SI,DEMO
LEA DI,STRNG
MOV CX,6
CLD
REPE CMPSB
jnz msg22
```

```
msg11:
mov AH,09H
mov DX,OFFSET msg1
int 21h
jmp exit
```

```
msg22:
mov AH,09H
mov DX,OFFSET msg2
int 21h
jmp exit
```

```
exit:int 3
```

CODE ENDS

END START

**Compilation /Running and Debugging steps:**

(As given in lab manual as an example of multiplication program on page no:5 of lab manual)

```

DOS
BOX
DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Pro...
A:\>tasm so12.asm
Turbo Assembler Version 3.0 Copyright (c) 1988, 1991 Borland International

Assembling file:    so12.asm
Error messages:    None
Warning messages:  None
Passes:            1
Remaining memory:  476k

A:\>tlink so12.obj
Turbo Link Version 3.0 Copyright (c) 1987, 1990 Borland International
Warning: No stack

A:\>tasm so12
Turbo Assembler Version 3.0 Copyright (c) 1988, 1991 Borland International

Assembling file:    so12.ASM
Error messages:    None
Warning messages:  None
Passes:            1
Remaining memory:  476k

A:\>

```

```

DOS
BOX
DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Pro...
A:\>debug so12.exe
-u
076E:0000 B86A07      MOV     AX,076A
076E:0003 8ED8        MOV     DS,AX
076E:0005 8EC0        MOV     ES,AX
076E:0007 BE0000      MOV     SI,0000
076E:000A BF0800      MOV     DI,0008
076E:000D B90600      MOV     CX,0006
076E:0010 FC         CLD
076E:0011 F3         REPZ
076E:0012 A6         CMPSB
076E:0013 750A        JNZ     001F
076E:0015 B409        MOV     AH,09
076E:0017 BA1000      MOV     DX,0010
076E:001A CD21        INT     21
076E:001C EB0B        JMP     0029
076E:001E 90         NOP
076E:001F B409        MOV     AH,09
-g
strings are same
AX=076A BX=0000 CX=0000 DX=0010 SP=0000 BP=0000 SI=0006 DI=000E
DS=076A ES=076A SS=0769 CS=076E IP=0029  NV UP EI PL ZR NA PE NC
076E:0029 CC         INT     3
-

```

```

DOS
DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Pro...
076E:001F B409          MOV     AH,09
-g
strings are same
AX=096A BX=0000 CX=0000 DX=0010 SP=0000 BP=0000 SI=0006 DI=000E
DS=076A ES=076A SS=0769 CS=076E IP=0029  NU UP EI PL ZR NA PE NC
076E:0029 CC          INT     3
-d ds:00
076A:0000 61 61 6E 61 6E 64 69 24-61 61 6E 61 6E 64 69 24  aaanandi$aanandi$
076A:0010 73 74 72 69 6E 67 73 20-61 72 65 20 73 61 6D 65  strings are same
076A:0020 24 73 74 72 69 6E 67 73-20 61 72 65 20 6E 6F 74  $strings are not
076A:0030 20 73 61 6D 65 24 00 00-00 00 00 00 00 00 00 00  same$.....
076A:0040 B8 6A 07 8E D8 8E C0 BE-00 00 BF 08 00 B9 06 00  .j.....
076A:0050 FC F3 A6 75 0A B4 09 BA-10 00 CD 21 EB 0B 90 B4  ...u.....!....
076A:0060 09 BA 21 00 CD 21 EB 01-90 CC 0C 00 26 89 36 1A  ..!...!.....&.6.
076A:0070 00 26 89 3E 18 00 80 CB-20 26 88 1E 05 00 26 89  .&.>.... &....&.
-d es:00
076A:0000 61 61 6E 61 6E 64 69 24-61 61 6E 61 6E 64 69 24  aaanandi$aanandi$
076A:0010 73 74 72 69 6E 67 73 20-61 72 65 20 73 61 6D 65  strings are same
076A:0020 24 73 74 72 69 6E 67 73-20 61 72 65 20 6E 6F 74  $strings are not
076A:0030 20 73 61 6D 65 24 00 00-00 00 00 00 00 00 00 00  same$.....
076A:0040 B8 6A 07 8E D8 8E C0 BE-00 00 BF 08 00 B9 06 00  .j.....
076A:0050 FC F3 A6 75 0A B4 09 BA-10 00 CD 21 EB 0B 90 B4  ...u.....!....
076A:0060 09 BA 21 00 CD 21 EB 01-90 CC 0C 00 26 89 36 1A  ..!...!.....&.6.
076A:0070 00 26 89 3E 18 00 80 CB-20 26 88 1E 05 00 26 89  .&.>.... &....&.

```

```

DOS
DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Pro...
A:\>debug sol2.exe
-g
strings are not same
AX=096A BX=0000 CX=0005 DX=0021 SP=0000 BP=0000 SI=0001 DI=0009
DS=076A ES=076A SS=0769 CS=076E IP=0029  NU UP EI NG NZ NA PO CY
076E:0029 CC          INT     3
-d ds:00
076A:0000 41 41 4E 41 4E 44 49 24-61 61 6E 61 6E 64 69 24  AANANDI$aanandi$
076A:0010 73 74 72 69 6E 67 73 20-61 72 65 20 73 61 6D 65  strings are same
076A:0020 24 73 74 72 69 6E 67 73-20 61 72 65 20 6E 6F 74  $strings are not
076A:0030 20 73 61 6D 65 24 00 00-00 00 00 00 00 00 00 00  same$.....
076A:0040 B8 6A 07 8E D8 8E C0 BE-00 00 BF 08 00 B9 06 00  .j.....
076A:0050 FC F3 A6 75 0A B4 09 BA-10 00 CD 21 EB 0B 90 B4  ...u.....!....
076A:0060 09 BA 21 00 CD 21 EB 01-90 CC 0C 00 26 89 36 1A  ..!...!.....&.6.
076A:0070 00 26 89 3E 18 00 80 CB-20 26 88 1E 05 00 26 89  .&.>.... &....&.
-d es:00
076A:0000 41 41 4E 41 4E 44 49 24-61 61 6E 61 6E 64 69 24  AANANDI$aanandi$
076A:0010 73 74 72 69 6E 67 73 20-61 72 65 20 73 61 6D 65  strings are same
076A:0020 24 73 74 72 69 6E 67 73-20 61 72 65 20 6E 6F 74  $strings are not
076A:0030 20 73 61 6D 65 24 00 00-00 00 00 00 00 00 00 00  same$.....
076A:0040 B8 6A 07 8E D8 8E C0 BE-00 00 BF 08 00 B9 06 00  .j.....
076A:0050 FC F3 A6 75 0A B4 09 BA-10 00 CD 21 EB 0B 90 B4  ...u.....!....
076A:0060 09 BA 21 00 CD 21 EB 01-90 CC 0C 00 26 89 36 1A  ..!...!.....&.6.
076A:0070 00 26 89 3E 18 00 80 CB-20 26 88 1E 05 00 26 89  .&.>.... &....&.

```



**Output:**

Screenshots of the output in both cases.

Case 1: Strings are same-

```

DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Pro...
076E:001F B409 MDU AH,09
-g
strings are same
AX=096A BX=0000 CX=0000 DX=0010 SP=0000 BP=0000 SI=0006 DI=000E
DS=076A ES=076A SS=0769 CS=076E IP=0029 NU UP EI PL ZR NA PE NC
076E:0029 CC INT 3
-d ds:00
076A:0000 61 61 6E 61 6E 64 69 24-61 61 6E 61 6E 64 69 24 aanandi$aanandi$
076A:0010 73 74 72 69 6E 67 73 20-61 72 65 20 73 61 6D 65 strings are same
076A:0020 24 73 74 72 69 6E 67 73-20 61 72 65 20 6E 6F 74 $strings are not
076A:0030 20 73 61 6D 65 24 00 00-00 00 00 00 00 00 00 00 same$.
076A:0040 B8 6A 07 8E D8 8E C0 BE-00 00 BF 08 00 B9 06 00 .j.
076A:0050 FC F3 A6 75 0A B4 09 BA-10 00 CD 21 EB 0B 90 B4 ...u...!...
076A:0060 09 BA 21 00 CD 21 EB 01-90 CC 0C 00 26 89 36 1A ..!..!...&.6.
076A:0070 00 26 89 3E 18 00 80 CB-20 26 88 1E 05 00 26 89 .&.>... &...&.

```

Case 2 :Strings are not same-

```

DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Pro...
A:\>debug so12.exe
-g
strings are not same
AX=096A BX=0000 CX=0005 DX=0021 SP=0000 BP=0000 SI=0001 DI=0009
DS=076A ES=076A SS=0769 CS=076E IP=0029 NU UP EI NG NZ NA PO CY
076E:0029 CC INT 3
-d ds:00
076A:0000 41 41 4E 41 4E 44 49 24-61 61 6E 61 6E 64 69 24 AANANDI$aanandi$
076A:0010 73 74 72 69 6E 67 73 20-61 72 65 20 73 61 6D 65 strings are same
076A:0020 24 73 74 72 69 6E 67 73-20 61 72 65 20 6E 6F 74 $strings are not
076A:0030 20 73 61 6D 65 24 00 00-00 00 00 00 00 00 00 00 same$.
076A:0040 B8 6A 07 8E D8 8E C0 BE-00 00 BF 08 00 B9 06 00 .j.
076A:0050 FC F3 A6 75 0A B4 09 BA-10 00 CD 21 EB 0B 90 B4 ...u...!...
076A:0060 09 BA 21 00 CD 21 EB 01-90 CC 0C 00 26 89 36 1A ..!..!...&.6.
076A:0070 00 26 89 3E 18 00 80 CB-20 26 88 1E 05 00 26 89 .&.>... &...&.

```

- Write an assembly language program which accepts a character and a string from the user and prints the position of the character in to the string if it is found, otherwise the message "NOT FOUND". For simplicity, enter the sting with length in single digit, that is less than or equal to 9.

**Rules for Operands:** Take your name as an input string and search one of the character from it.

**Write your code here:**

DATA SEGMENT

OP1 DB "ENTER A STRING :\$"

```
STR_BUFF DB 15,16 DUP(0)
OP2 DB 0Dh,0Ah,"ENTER A CHARACTER :$"
MESS1 DB 0Dh,0Ah,"CHARACTER FOUND AT THE POSITION :$"
MESS2 DB 0Dh,0Ah,"CHARACTER NOT FOUND$"
DATA ENDS
```

#### CODE SEGMENT

```
ASSUME CS:CODE , DS:DATA ,ES:DATA
```

```
START : MOV AX,DATA
```

```
        MOV DS,AX
```

```
        MOV ES,AX
```

```
        MOV AH,09h
```

```
        LEA DX,OP1          ;DISPLAY OP1 MESSAGE
```

```
        INT 21h
```

```
        MOV AH,0Ah
```

```
        LEA DX,STR_BUFF      ;GET A STRING IN STR_BUFF
```

```
        INT 21h
```

```
        MOV AH,09h
```

```
        LEA DX,OP2          ;DISPLAY OP2 MESSAGE
```

```
        INT 21h
```

```
        MOV AH,01h
```

```
        INT 21h            ;GET A CHARACTER
```

```
        MOV DI,OFFSET STR_BUFF+1
```

```
        MOV CX,00
```

```
        MOV CL,BYTE PTR[DI]
```



```
    INC DI
    MOV BX,DI
    CLD
REPNE SCASB
    JNZ NOTFOUND

    MOV AH,9
    LEA DX,MESS1      ;CHARACTER FOUND
    INT 21h

    SUB DI,BX
    MOV DX,DI
    ADD DL,'0'
    MOV AH,2
    INT 21h
    JMP EXIT

NOTFOUND: MOV AH,09h
          LEA DX,MESS2      ;CHARACTER NOT FOUND
          INT 21h

EXIT : MOV AX,4C00h
       INT 21h

CODE ENDS

END START
```

**Compilation /Running and Debugging steps:**

(As given in lab manual as an example of multiplication program on page no:5 of lab manual)

```
DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Pro...
A:\>tasm so13.asm
Turbo Assembler Version 3.0 Copyright (c) 1988, 1991 Borland International

Assembling file:    so13.asm
Error messages:     None
Warning messages:   None
Passes:             1
Remaining memory:   475k

A:\>tlink so13.obj
Turbo Link Version 3.0 Copyright (c) 1987, 1990 Borland International
Warning: No stack

A:\>tasm so13
Turbo Assembler Version 3.0 Copyright (c) 1988, 1991 Borland International

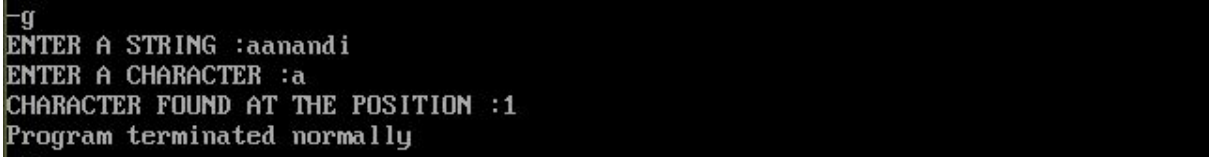
Assembling file:    so13.ASM
Error messages:     None
Warning messages:   None
Passes:             1
Remaining memory:   475k
```

```
DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Pro...
-u
0772:0000 B86A07      MOV     AX,076A
0772:0003 8ED8        MOV     DS,AX
0772:0005 BEC0        MOV     ES,AX
0772:0007 B409        MOV     AH,09
0772:0009 BA0000      MOV     DX,0000
0772:000C CD21        INT     21
0772:000E B40A        MOV     AH,0A
0772:0010 BA1100      MOV     DX,0011
0772:0013 CD21        INT     21
0772:0015 B409        MOV     AH,09
0772:0017 BA2200      MOV     DX,0022
0772:001A CD21        INT     21
0772:001C B401        MOV     AH,01
0772:001E CD21        INT     21
-g
ENTER A STRING :aanandi
ENTER A CHARACTER :a
CHARACTER FOUND AT THE POSITION :1
Program terminated normally
-g
ENTER A STRING :aanandi
ENTER A CHARACTER :z
CHARACTER NOT FOUND
A:\>
```

**Output:**


Screenshots of the output in both cases.

1. Character FOUND



```
-g
ENTER A STRING :aanandi
ENTER A CHARACTER :a
CHARACTER FOUND AT THE POSITION :1
Program terminated normally
```

2. Character NOT FOUND



```
-g
ENTER A STRING :aanandi
ENTER A CHARACTER :z
CHARACTER NOT FOUND
A:\>
```