Date: 25-02-2022 Exp. 05 Logical Operations & Number conversions

DATE : 3-03-2022 EXP:6 LENGTH OF STRING AND REVERSE OF STRING

**Aim:**

* To find the length of a given string
* To reverse a given string

**Tool Used:**

Assembler - MASM 611

**LENGTH OF A GIVEN STRING**

**ALGORITHM**

* **Declare and initialize the string in data segment**
* **Move the data to ds register via ax register.**
* **Load the starting address of str in si.**
* **Store ‘$’ in al register to detect the end of a string.**
* **Move the si register to the last element of the string: o Compare [si] with ‘$’ o If its equal, break out of loop If its not equal, increment si Repeat**

**PROGRAMME**

DATA SEGMENT

    STRING1 DB  "I AM PRATEEK$"

    STRING2 EQU "$"

DATA ENDS

CODE SEGMENT

          ASSUME CS:CODE ,DS:DATA

    START:

          MOV    AX,DATA

          MOV    DS,AX

          LEA    SI,STRING1

          MOV    BL,STRING2

          XOR    CX,CX

    LOOP1:

          MOV    AL,[SI]

          INC    CX

          INC    SI

          CMP    AL,BL

          JNZ    LOOP1

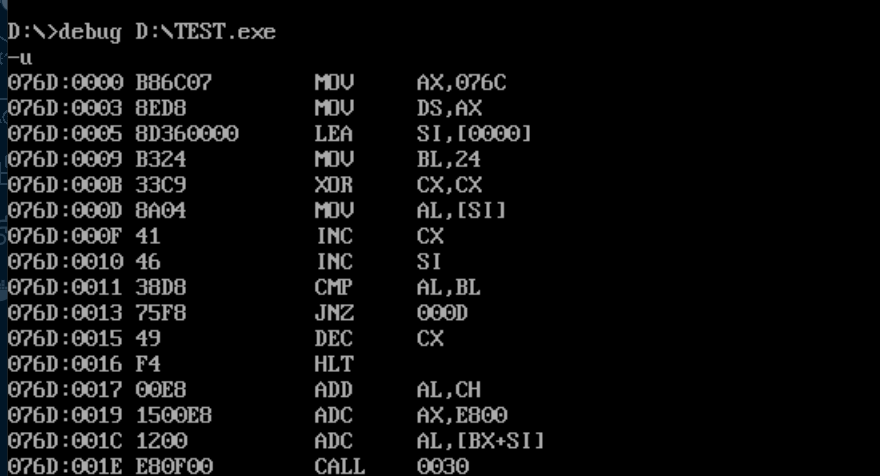
          DEC    CX

          HLT

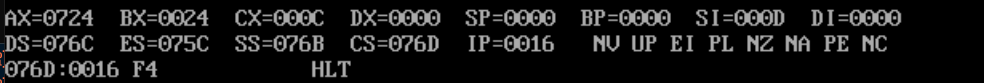
CODE ENDS

END START

**Register/ Memory Contents for I/O:**

****

**SNAPSHOT OF OUTPUT**

****

**REVERSING A GIVEN STRING**

**ALGORITHM**

STEP 1 : WRITE THE DATA SEGMENT  
  
STEP 2 : MOVE DATA TO AX  
  
STEP 3 : MOVE AX TO DS  
  
STEP 4 : LOAD EFFECTIVE ADDRESS OF MSG1 TO DX  
  
STEP 5 : PRINT MESSAGE  
  
STEP 6 : MOVE 0DH TO DL  
  
STEP 7 : LOAD EFFECTIVE ADDRESS OF ST1 TO S1  
  
STEP 8 : MOVE 00H TO CL  
  
STEP 9 : INPUT A CHARACTER  
  
STEP 10 : COMPARE IF IT IS A LINE FEED  
  
STEP 11 : JUMP IF ZERO TO STEP 18  
  
STEP 12 : INCREMENT CL  
  
STEP 13 : INCREMENT SI  
  
STEP 14 : MOVE AL TO THE CONTENT OF SI  
  
STEP 15 : INCREMENT SI  
  
STEP 16 : INCREMENT CL  
  
STEP 17 : JUMP TO STEP 9  
  
STEP 18 : MOVE 0AH TO DL  
  
STEP 19 : MOVE 0DH TO DL  
  
STEP 20 : LOAD EFFECTIVE ADDRESS OF MSG2 TO DX  
  
STEP 21 : PRINT THE MESSAGE  
  
STEP 22 : MOVE THE CONTENT OF SI TO DL  
  
STEP 23 : PRINT THE CHARACTER WHERE VALUE IN DL  
  
STEP 24 : DECREMENT SI  
  
STEP 25 : DECREMENT CL  
  
STEP 26 : COMPARE 00H TO CL  
  
STEP 27 : JUMP IF NOT ZERO TO STEP 21  
  
STEP 28 : TERMINATE THE PROGRAM  
  
STEP 29 : STOP

**PROGRAMME**

DATA SEGMENT

    STRING1 DB  "PRATEEK$"

    STRING2 EQU "$"

    PAL     DB  00H

DATA ENDS

EXTRA SEGMENT

    ARR2  DB 07H DUP(?)

EXTRA ENDS

CODE SEGMENT

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

    START:

          MOV    AX,DATA

          MOV    DS,AX

          MOV    AX,EXTRA

          MOV    ES,AX

          LEA    SI,STRING1

          MOV    BL,STRING2

          XOR    CX,CX

    LOOP1:

          MOV    AL,[SI]

          INC    CX

          INC    SI

          CMP    AL,BL

          JNZ    LOOP1

          DEC    CX

          DEC    CX

          LEA    SI,STRING1

          LEA    DI,ARR2

    LOOP2:

          INC    DI

          DEC    CX

          CMP    CX,00H

          JNZ    LOOP2

    LOOP3:

          CLD

          LODSB

          STD

          STOSB

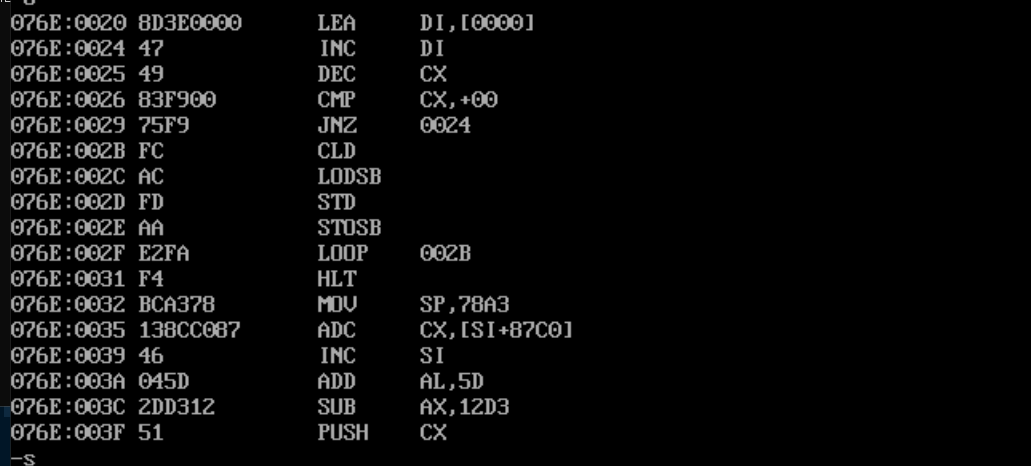
          LOOP   LOOP3

          HLT

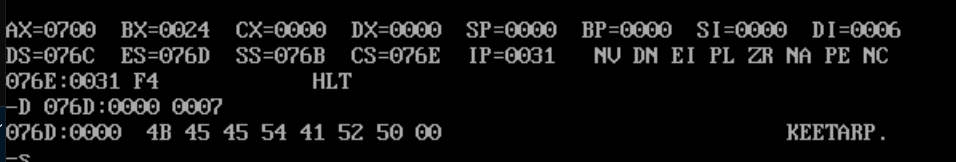
CODE ENDS

END START

**Register/ Memory Contents for I/O:**

****

**SNAPSHOT OF OUTPUT**

****

**Result:**

**Thus the program was executed successfully**.