CSE 2006

MICROPROCESSOR AND INTERFACING

Lab FAT Exam

L11+L12 | SJT516

FALL SEMESTER 2021-22

by

SHARADINDU ADHIKARI

19BCE2105

Problem statement:

(a) Write and verify 8086 assembly language program that will perform following string operation for a string : "This is a microprocessor and interfacing lab exam"
i. Calculate length of a string
ii. Count number of spaces in a string
iii. Reverse the given string and print the reversed string
(b) Write only an ALP and calculate delay to control the different direction of rotation of stepper motor. Stepper motor has the following specification
i. Stepper motor has 4 windings
ii. Speed =12 rpm
iii. No of teeth= 300

Part (a):

AIM: The aim of this particular experiment is to write an ALP program to that willperform following string operation for a string: "This is a microprocessor and interfacing lab exam"

i. Calculate length of a string
ii. Count the number of spaces in a string
iii. Reverse the given string and print the reversed string

Tool Requirement: emu8086 (it is a free emulator which provides emulating of old8086 processors)

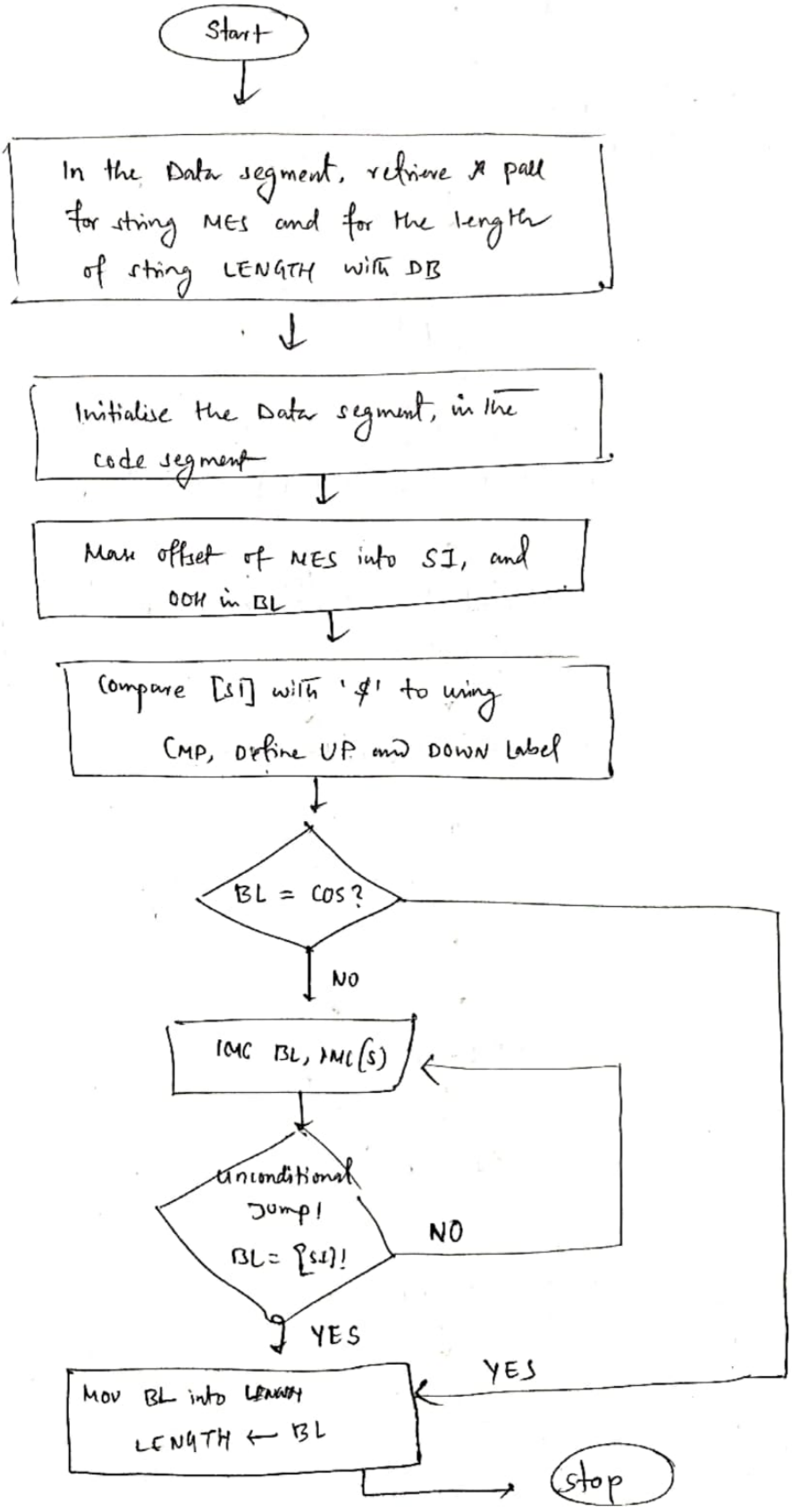
Procedure:

1. First, we write the code that is in assembly language so we save the file using an .asm extension to the file.
2. After writing the assembly language code we click on the compile button onthe emu8086 screen.
3. After the compilation is done the .asm code will be changed to .bin code andthe emulator will ask you to save the code somewhere.
4. After that you will click on the emulate button which will call the linker andlocator to convert the .obj file to .exe file which is an executable file.
5. Then a window will open showing all the registers and other information showing in it. Now you can either run a single step at a time or you can run thewhole program in one go.
6. After running the whole program, we can check the registers as well as the memory for the correct values. In this way we can know if your program isworking properly or not.

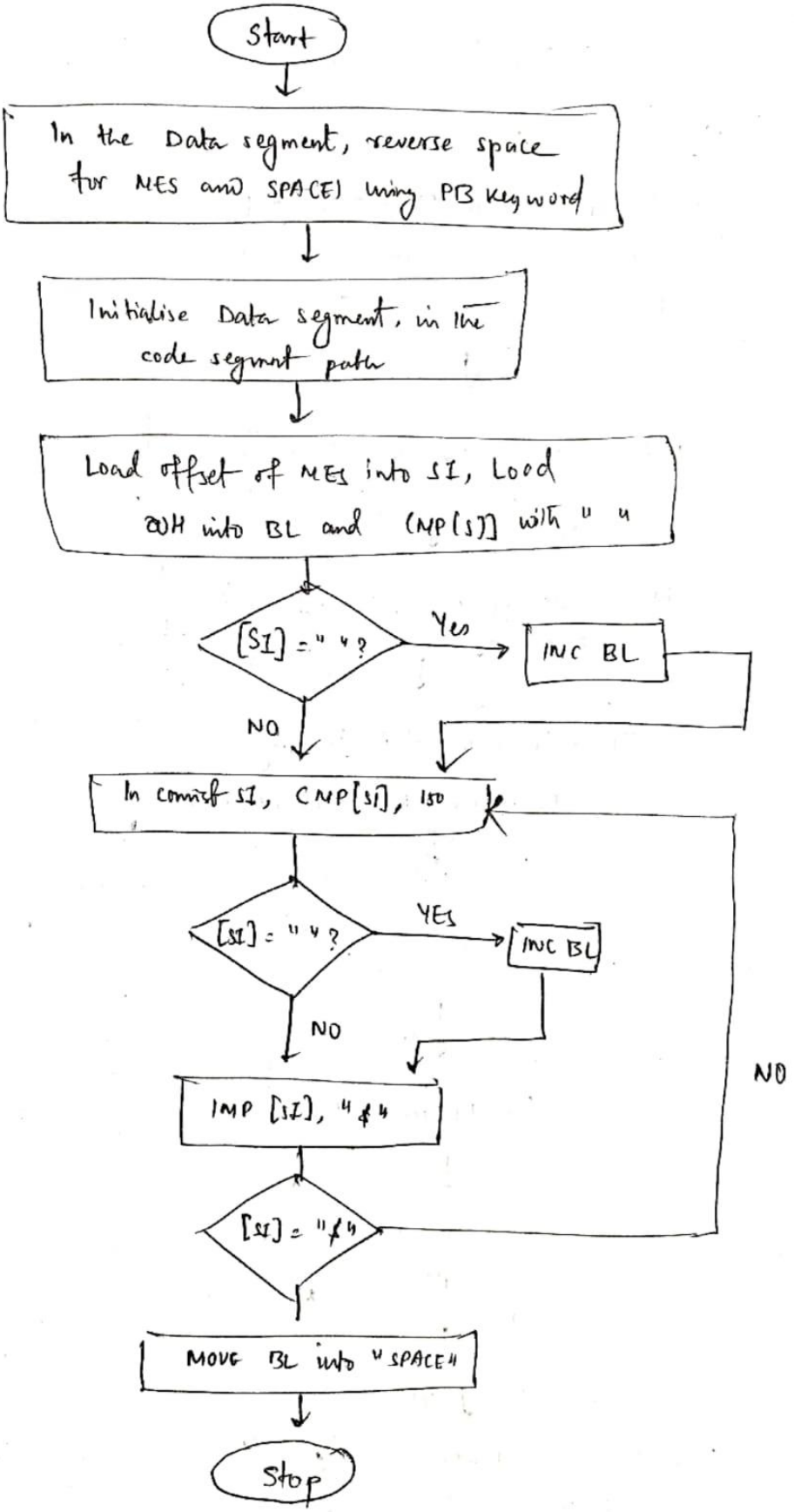
sharadindu.adhikari2019@vitstudent.ac.in

FLOWCHARTS:

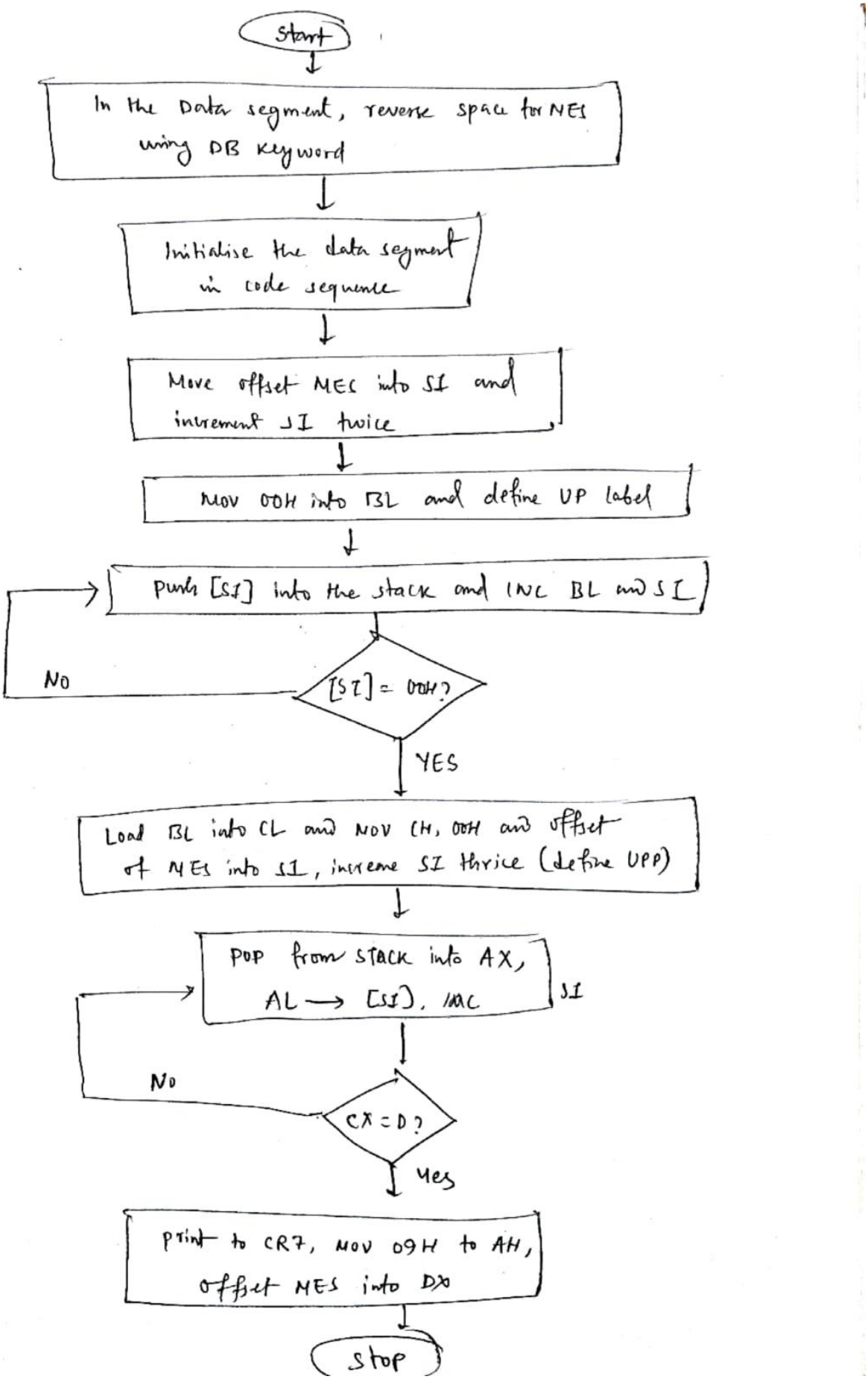
Length of string:



No. of spaces:


Start
In the Data segment, reserve space for MES and SPACE1 using PB keyword
Initialise Data segment, in the code segment path
Load offset of MES into SI, Load 00H into BL and CMP[SI] with " "
[SI] = " "?
Yes
INC BL
NO
In comstr SI, CMP[SI], 150
[SI] = " "?
YES
INC BL
NO
IMP [SI], "$"
[SI] = "$"
NO
MOVE BL into "SPACE1"
Stop

Reverse the string:



INPUT:

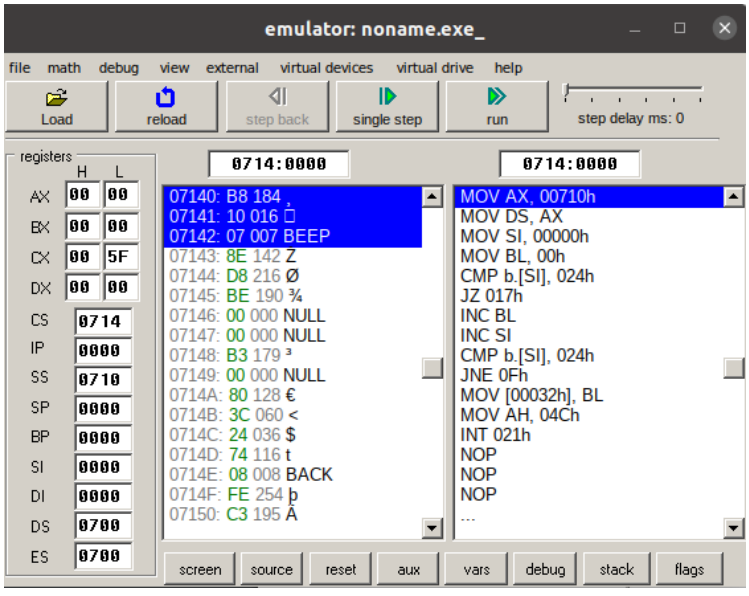
(i) Length of a string

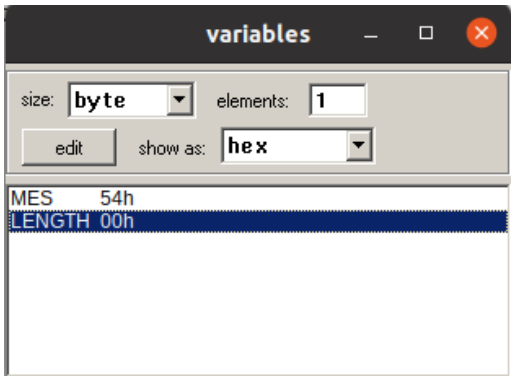
```
; Sharadindu Adhikari
; 19BCE2105
; Micro Lab FAT


DATA SEGMENT
  MES DB "This is a microprocessor and interfacing lab exam",
  "$" LENGTH DB DUP(0)
DATA ENDS

CODE SEGMENT
  ASSUME CS:CODE, DS:DATA

  START:
    ; data segment MOV
    AX, DATA MOV DS, AX

    MOV SI, OFFSET MES MOV

    BL, 00H
    CMP [SI], "$"
    JZ DOWN

    UP:
      INC BL

      INC SI
      CMP [SI], "$" JNZ
      UP

    DOWN:
      MOV LENGTH, BL

    ; software interrupt
    MOV AH, 04CH INT 21H
CODE ENDS END
START
```

```
; Sharadindu Adhikari
; 19BCE2105
; Micro Lab FAT

DATA SEGMENT
MES DB "This is a microprocessor and interfacing lab exam",
"$" LENGTH DB DUP(0)
DATA ENDS
CODE SEGMENT
ASSUME CS:CODE, DS:DATA
START:
; data segment MOV
AX, DATA MOV DS, AX
MOV SI, OFFSET MES MOV
BL, 00H
CMP [SI], "$"
JZ DOWN
UP:
INC BL
INC SI
CMP [SI], "$" JNZ
UP
DOWN:
MOV LENGTH, BL
; software interrupt
MOV AH, 04CH INT 21H
CODE ENDS END
START
```

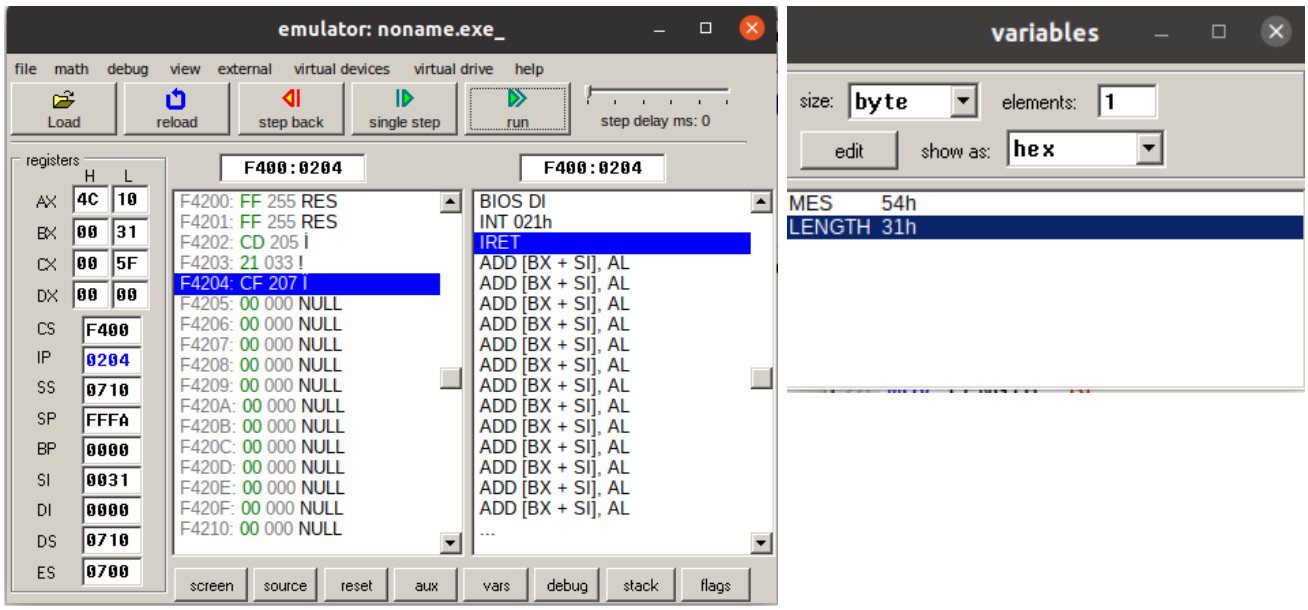
Output:

Before execution:





After execution:



(ii) No. of spaces in string

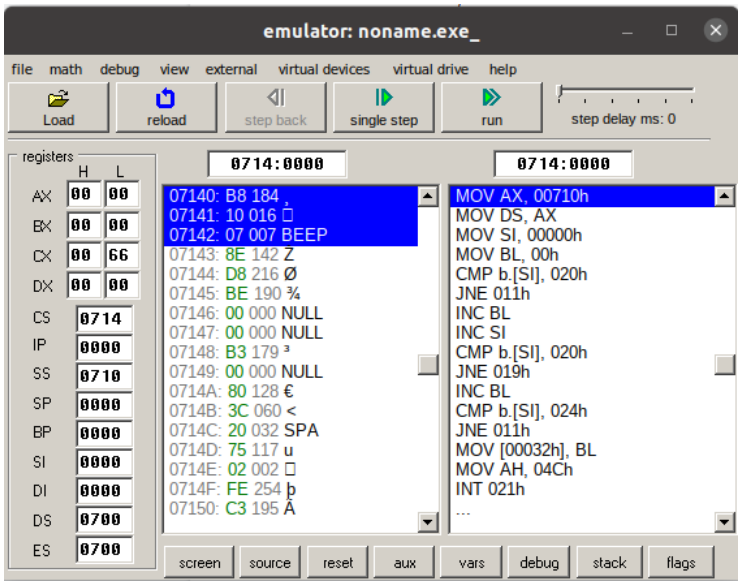
```
; Sharadindu Adhikari
; 19BCE2105
; Micro Lab FAT


DATA SEGMENT
  MES DB "This is a microprocessor and interfacing lab exam",
  "$" SPACES DB DUP(0)
DATA ENDS

CODE SEGMENT
  ASSUME CS:CODE, DS:DATA

  START:
    ; data segment MOV
    AX, DATA MOV DS, AX

    MOV SI, OFFSET MES MOV

    BL, 00H
    CMP [SI], " "
    JNZ UP INC BL

    UP:
      INC SI
      CMP [SI], " "
      JNZ DOWN INC BL
      DOWN:
        CMP [SI], "$" JNZ
```

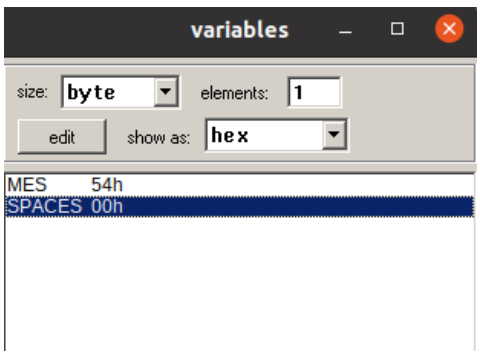
```
        UP

    MOV SPACES, BL

    ; software interrupt
    MOV AH, 04CH INT 21H
CODE ENDS END
START
```

```
; Sharadindu Adhikari
; 19BCE2105
; Micro Lab FAT


; Sharadindu Adhikari
; 19BCE2105
; Micro Lab FAT
DATA SEGMENT
MES DB "This is a microprocessor and interfacing lab exam",
"$" SPACES DB DUP(0)
DATA ENDS
CODE SEGMENT
ASSUME CS:CODE, DS:DATA
START:
; data segment MOV
AX, DATA MOV DS, AX
MOV SI, OFFSET MES MOV
BL, 00H
CMP [SI], " "
JNZ UP INC BL
UP:
INC SI
CMP [SI], " "
JNZ DOWN INC BL
DOWN:
CMP [SI], "$" JNZ
UP
MOV SPACES, BL
; software interrupt
MOV AH, 04CH INT 21H
CODE ENDS END
START
```

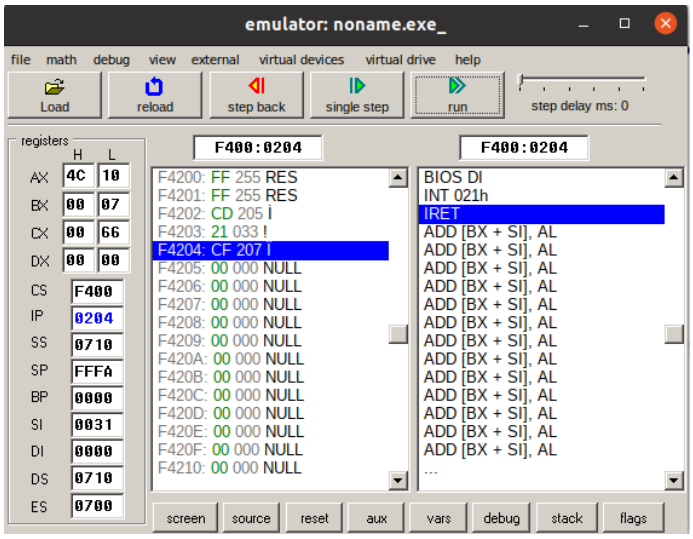
OUTPUT:

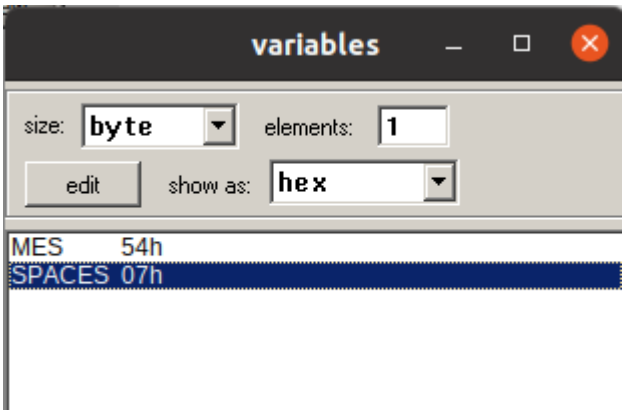
Before execution:





After execution:





(iii) Reverse a string

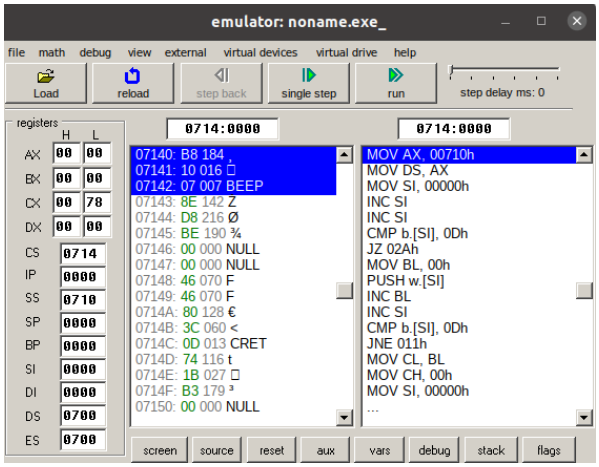
```
; Sharadindu Adhikari
; 19BCE2105
; Micro Lab FAT

 DATA SEGMENT
   MES DB 0DH, 0AH, "This is a microprocessor and interfacing lab exam", 0DH,
   0AH, "$" SPACES DB DUP(0)
 DATA ENDS

 CODE SEGMENT
   ASSUME CS:CODE, DS:DATA

   START:
     ; data segment MOV
     AX, DATA MOV DS, AX

     MOV SI, OFFSET MES INC SI
     INC SI

     CMP [SI], 0DH JZ
     PRINT

     MOV BL, 00H UP:
       PUSH [SI] INC
       BL INC SI
       CMP [SI], 0DH JNZ
       UP
```

```
    MOV CL, BL MOV
    CH, 00H
    MOV SI, OFFSET MES INC SI
    INC SI

    UPP:
      POP AX MOV [SI],
      AL INC SI

      LOOP UPP PRINT:
      MOV CX, 04H;

      MOV AH, 09H
      MOV DX, OFFSET MES
      INT 21H

    ; software interrupt
    MOV AH, 04CH INT 21H
CODE ENDS END
START
```

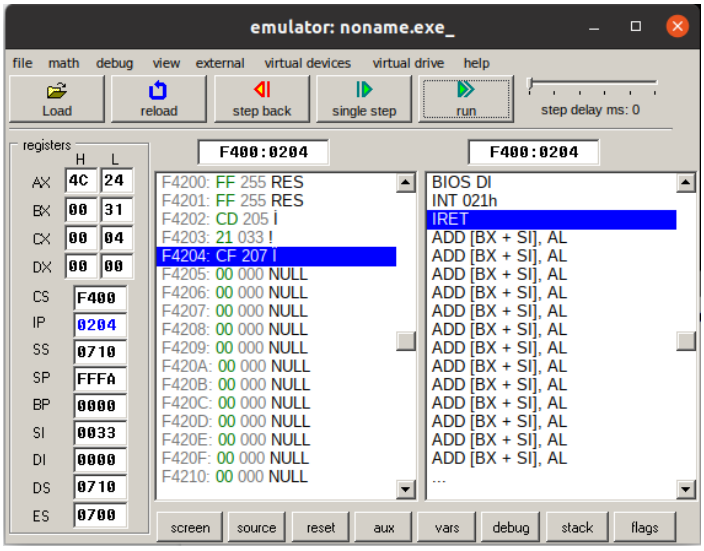
```
; Sharadindu Adhikari
; 19BCE2105
; Micro Lab FAT

DATA SEGMENT
MES DB 0DH, 0AH, "This is a microprocessor and interfacing lab exam", 0DH, 0AH, "$" SPACES DB DUP(0)
DATA ENDS

CODE SEGMENT
ASSUME CS:CODE, DS:DATA

START:
; data segment MOV AX, DATA MOV DS, AX

MOV SI, OFFSET MES INC SI
INC SI

CMP [SI], 0DH JZ PRINT

MOV BL, 00H UP:
PUSH [SI] INC BL INC SI
CMP [SI], 0DH JNZ UP
MOV CL, BL MOV CH, 00H
MOV SI, OFFSET MES INC SI
INC SI

UPP:
POP AX MOV [SI], AL INC SI
LOOP UPP PRINT:
MOV CX, 04H;

MOV AH, 09H
MOV DX, OFFSET MES
INT 21H

; software interrupt MOV AH, 04CH INT 21H
CODE ENDS END START

```

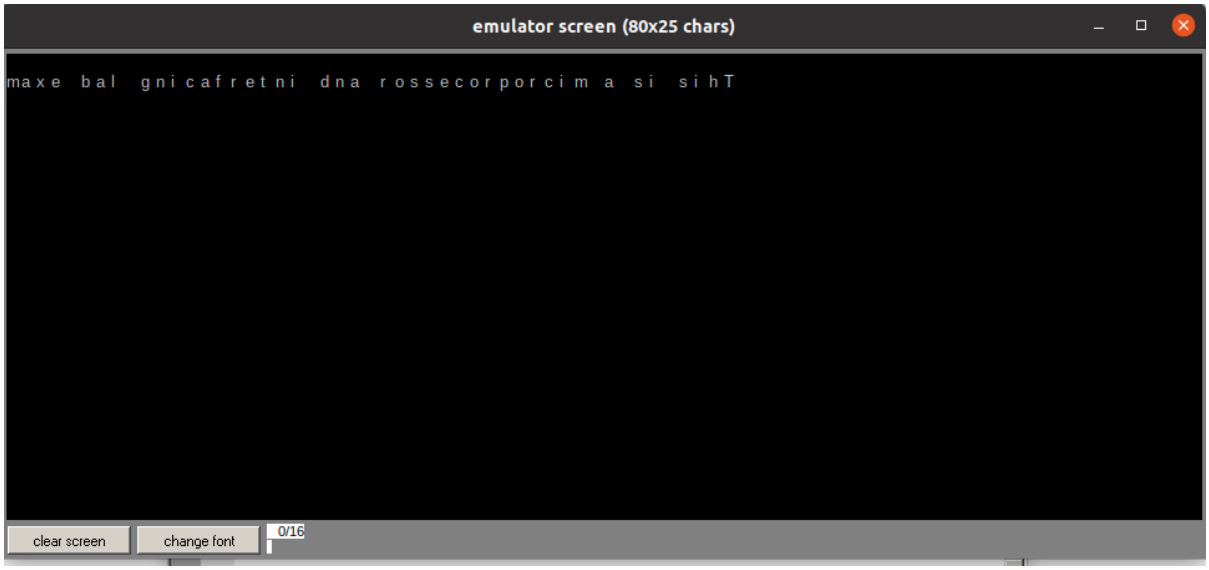
OUTPUT:

Before execution:



After execution:





INFERENCE:

From the emulator execution, we find that the length of the string is 31H which is 49 in decimal. Hence the ALP is working. Also the spaces in the string are equal to 7, which is also true. Also, the CRT screen shows the reverse of the string, which is also as expected.

It is clearly observable from the screenshots appended that the string operations are working correctly as required by the specification in the problem statement.

Part (b)

19BCE2105

Given speed = 12 rpm

No. of teeth = 300

∴ In 60s, there are = 12 rotations

∴ In 1s, there are = $\frac{12}{60} = \frac{1}{5}$ rotations

Also,

1 rotation = 300 teeth

∴ $\frac{1}{5}$ rotation = $300 \times \frac{1}{5}$ = 60 teeth

ALP program:

```
DATA SEGMENT

PORT A DB ~~0FFH~~ 0FCH

PORT B DB 0FDH

PORT C DB 0FEH

CWR DB 0FFH

DATA ENDS

CODE SEGMENT:
```

(P.T.O.)

sharadindu.adhikari2019@vitstudent.ac.in

~~START:~~

```
ASSUME   CS: CODE, DS: DATA
START:   MOV  AX, DMA
         MOV  DS, AX
         MOV  AL, 060H
         MOV  DX, LWR
         MOV  DX, AL
         MOV  AL, 033H
         MOV  AX, 00H
; for clockwise rotation
         MOV  CX, 04H  ; for 4 rotation on windings
         MOV  DX, PORT A
UP 1 :   OUT  DX, AL
         CALL DELAY
         ROR  AL, 01
         LOOP UP1
; for anticlockwise rotation
```

```
        MOV   AL, 033H
        MOV   AH, 00H
        MOV   CX, 64H

UP2:    OUT   DX, AL
        CALL  DELAY
        RUL   AL, 01
        LOOP  UP2
```