

AIDS MICROPROCESSOR LAB S21 BATCH (2023-24)

Experiment 7(a) Title: Assembly language programming based on String operation.

Name of student: Meet Raut Class Roll Number: 2201084

Date of Performance: 01/04/2024

Batch: S2-1 Timing: 3:00-5:00 Date of Submission: 01/04/2024

Assembly language code

DATA_SEG SEGMENT

STR1 DB 23H,34H,45H,65H,76H,84H,12H,54H,65H,22H

STR2 DB 10 DUP(0)

DATA_SEG ENDS

CODE_SEG SEGMENT

ASSUME CS:CODE_SEG , DS:DATA_SEG , ES:DATA_SEG

START :

MOV AX,DATA_SEG #INITIALISE THE DATA SEGMENT REGISTER

MOV DS,AX

MOV ES,AX

MOV CX,10

MOV SI,OFFSET STR1 #LOAD POINTER TO STR1

MOV DI,OFFSET STR2 #LOAD POINTER TO STR2

CLD #CLEAR DIRECTION FLAG

REP MOVSB #MOVE ONE BYTE

MOV AH,4CH #REQUEST TO TERMINATE

INT 21H #EXIT TO DOS

CODE_SEG ENDS

END START

Result:

```

File Edit View Run Breakpoints Data Options Window Help READY
Module: p7a_17 File: p7a_17.asm 30 1

ASSUME CS:CODE_SEG , DS:DATA_SEG , ES:DATA_SEG
• START : MOV AX,DATA_SEG ;INITIALISE THE DATA SEGMENT REGISTER
•         MOV DS,AX
•         MOV ES,AX
•         MOV CX,10
•         MOV SI,OFFSET STR1 ;LOAD POINTER TO STR1
•         MOV DI,OFFSET STR2 ;LOAD POINTER TO STR2
•         CLD
•         REP MO
ds:0000 23 34 45 65 76 84 12 54 #4Eevä#T
ds:0008 65 22 23 34 45 65 76 84 e"#4Eevä
ds:0010 12 54 65 22 00 00 00 00 #Te"
•         MOV AH
ds:0018 00 00 00 00 00 00 00 00
•         INT 21

CODE_SEG ENDS

```

Experiment 7(b) Title: Assembly language programming based on String operation.

Name of student: Meet Raut **Class Roll Number: 2201084**

Date of Performance: 01/04/2024

Date of Submission: 01/04/2024

Assembly language code

DATA_SEG SEGMENT

STR1 DB 23H,34H,45H,65H,76H,84H,12H,54H,65H,22H

DATA_SEG ENDS

CODE_SEG SEGMENT

ASSUME CS:CODE_SEG, DS:DATA_SEG, ES:DATA_SEG

START :

```
MOV AX,DATA_SEG
```

#INITIALISE THE DATA SEGMENT REGISTER

MOV DS,AX

MOV ES,AX

```
MOV CX,10
```

```
MOV SI,OFFSET STR1
```

```
#LOAD POINTER TO STR1
```

ADD SI,9

#SI WILL POINT AT THE LAST INDEX

MOV DI,SI

#DI WILL POINT AT THE LAST INDEX

ADD DI,5

#DI WILL POINT TO 5 INDEX AFTER LAST INDEX

STD

REP MOVSB

MOV AH,4CH

#REQUEST TO TERMINATE

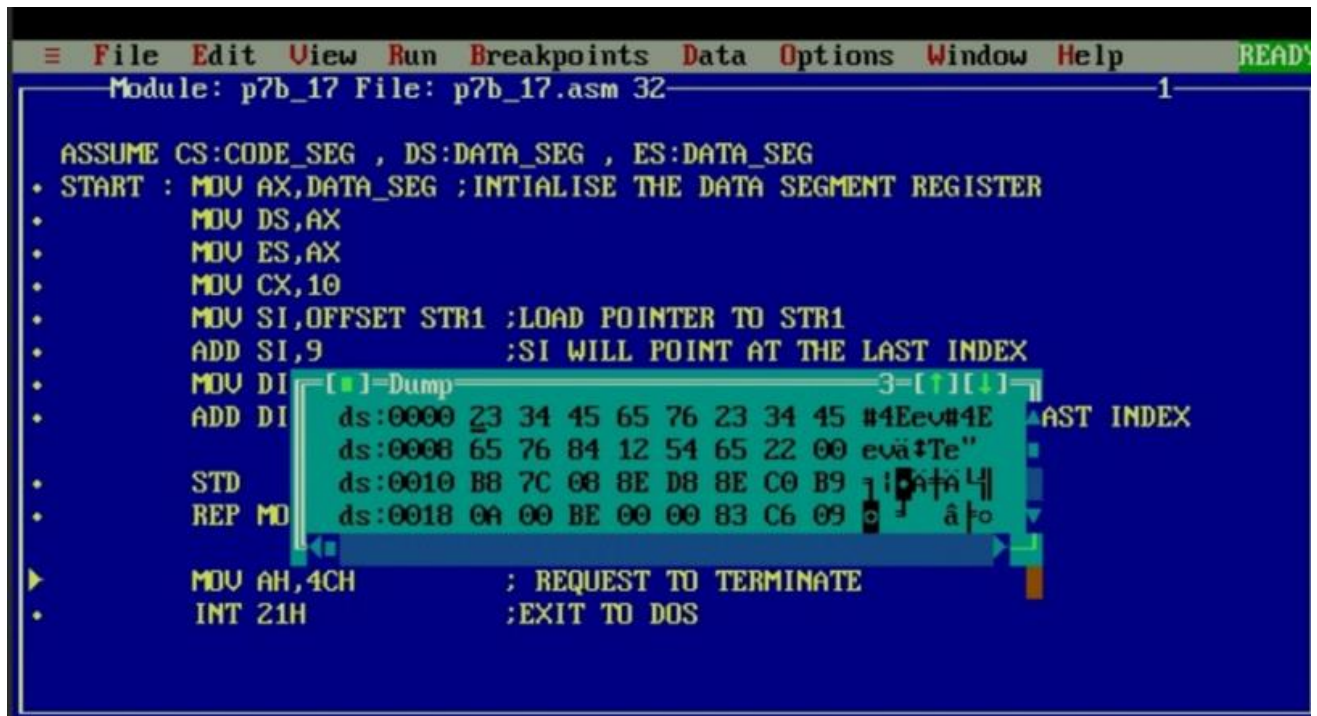
INT 21H

#EXIT TO DOS

CODE_SEG ENDS

END START

Result:



The screenshot shows a DOS assembly program running in a debugger. The menu bar includes File, Edit, View, Run, Breakpoints, Data, Options, Window, and Help. The status bar indicates the module is p7b_17, the file is p7b_17.asm, and the current instruction is at address 32. The program code is as follows:

```
ASSUME CS:CODE_SEG , DS:DATA_SEG , ES:DATA_SEG
* START : MOV AX,DATA_SEG ;INITIALISE THE DATA SEGMENT REGISTER
*         MOV DS,AX
*         MOV ES,AX
*         MOV CX,10
*         MOV SI,OFFSET STR1 ;LOAD POINTER TO STR1
*         ADD SI,9           ;SI WILL POINT AT THE LAST INDEX
*         MOV DI
*         ADD DI
*         STD
*         REP MD
*         MOV AH,4CH         ; REQUEST TO TERMINATE
*         INT 21H           ;EXIT TO DOS
```

A memory dump window is open, showing the contents of the data segment (ds) starting at address 0000. The dump is organized into columns of 16-bit values (hex and decimal) and a column of ASCII characters. The first row shows the values 23, 34, 45, 65, 76, 23, 34, 45, which correspond to the ASCII string "Eev#4E". The second row shows 65, 76, 84, 12, 54, 65, 22, 00, which correspond to the ASCII string "evä#Te". The third row shows BB, 7C, 08, 8E, D8, 8E, C0, B9, which correspond to the ASCII string "iA#A4". The fourth row shows 0A, 00, BE, 00, 00, 83, C6, 09, which correspond to the ASCII string "o âto". The dump window has a title bar that says "[]=Dump" and a status bar that says "3-[↑][↓]". The label "LAST INDEX" is visible on the right side of the dump window.

CONCLUSION: LO 2, LO 3 mapped.
