Rajshahi University of Engineering & Technology

Lab Report No.05

Course No: CSE 3110

Submitted To:

Sadia Zaman Mishu
Assistant Professor

Department of Computer Science and Engineering , RUET.

Submitted By:

Khan Fashee Monowar
Roll: 1503084
Department of Computer Science and Engineering, RUET.

Experiment Number: 05

Name of the Experiment: Assembly Programs to display the capital letter entered that comes first alphabetically and the one that comes last.

Theory:

The CMP instruction compares two operands. It is generally used in conditional execution. This instruction basically subtracts one operand from the other for comparing whether the operands are equal or not. It does not disturb the destination or source operands. It is used along with the conditional jump instruction for decision making.
CMP destination, source
CMP compares two numeric data fields. The destination operand could be either in register or in memory. The source operand could be a constant (immediate) data, register or memory.
The JMP instruction provides a label name where the flow of control is transferred immediately. The syntax of the JMP instruction is -
JMP label.

Conditional execution in assembly language is accomplished by several looping and branching instructions. These instructions can change the flow of control in a program. Conditional execution is observed in two scenarios -

Unconditional jump: This is performed by the JMP instruction. Conditional execution often involves a transfer of control to the address of an instruction that does not follow the currently executing instruction. Transfer of control may be forward, to execute a new set of instructions or backward, to re-execute the same steps.

Conditional jump: This is performed by a set of jump instructions j<condition> depending upon the condition. The conditional instructions transfer the control by breaking the sequential flow and they do it by changing the offset value in IP.

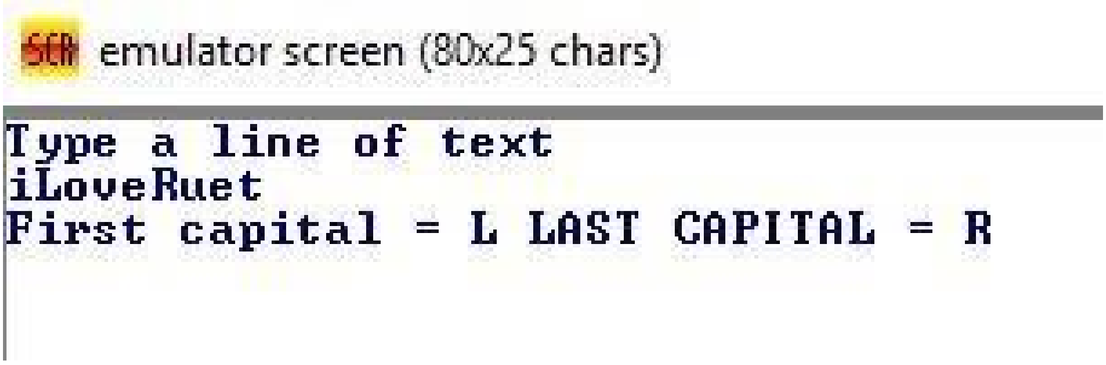
Code:

```
.MODEL SMALL
.STACK 100H
.DATA
PROMPT DB 'Type a line of text',0DH,0AH,'$'
NOCAP_MSG DB 0DH,0AH,'No capital $'
CAP_MSG DB 0DH,0AH,'First capital = '
FIRST DB ']',0DH,0AH
DB 'Last capital = '
LAST DB '@ $'
```

```
.CODE
MAIN PROC
   MOV AH, 06h
   XOR AL, AL
   XOR CX, CX
   MOV DX, 184FH      ;screen background color change
   MOV BH, 0F1h
   INT 10H
MOV AX,@DATA
MOV DS,AX
MOV AH,9
LEA DX,PROMPT
INT 21H
MOV AH,1
INT 21H

WHILE_:
CMP AL,0DH
JE END_WHILE
CMP AL,'A'
JNGE END_IF
CMP AL,'Z'
JNLE END_IF
CMP AL,FIRST
JNL CHECK_LAST
MOV FIRST,ALCHECK_LAST:
CMP AL,LAST
JNG END_IF
MOV LAST,AL
END_IF:

INT 21H
JMP WHILE_
END_WHILE:
MOV AH,9
CMP FIRST,']'
JNE CAPS
LEA DX,NOCAP_MSG
JMP DISPLAY
CAPS:
LEA DX,CAP_MSG
DISPLAY:
```

```
INT 21H
MOV AH,4CH
INT 21H

MAIN ENDP
END MAIN
```

Result:



```
Type a line of text
iLoveRuet
First capital = L LAST CAPITAL = R
```