KHULNA UNIVERSITY OF ENGINEERING AND TECHNOLOGY, KUET

SESSIONAL REPORT

Course No: CSE 2204

Department of: Computer Science and Engineering

Experiment No: **10**

Name of the Experiment To develop an assembly program that checks whether a string is a palindrome or not

Remarks

Date of Performance: 19.05.21

Date of Submission: 24.05.21

Name: Rifat Arefin

Roll: 18070117

Year: 2nd

Semester: 2nd

No. of experiment : 10

Name of experiment: To develop a program that checks whether a string is a palindrome or not in assembly language.

Objectives:

1. To practice assembly program with string data to more clear idea about string
2. To learn about palindrome string.
3. To develop a program that checks a string data is palindrome or not.

Introduction:

A string is a data type used in program, such as an integers and floating point unit but is used to represent next rather than numbers

A palindrome is a string that is the same

in forward or revese direction. For example, "dad" is the same in forward or reverse direction.

Another example is "aibohphobia", which literally same in forward and reverse direction too.

One approach to check palindrome is, iterate through the string till middle of the string and compare a character from back and front.

Apparatus required: emu 8086, laptop.

Methodology:

Code:

```
org 100h
mov ax, len  ; copy the length of 'str' to ax
mov bx, 02h  ; copy 02h to bx register
```

```
div bx ; divide the value stores in ax by 02h
mov cx, ax ; copy the value of ax to cx.

lea si, str ; store the address to the str to si
lea di, str ; store the address of str to di
add di, len ; add di with len
dec di ; decrement the value of di to
         point the last character of the
         string

Lup:
    mov bl, [si] ; copy the value stored in memory
                   pointed by si to bl
    mov bh, [di] ; copy the value stored in memory
                   pointed by di to bh

    cmp bl, bh
    jne not_equal ; if the values of bl and bh
                    then go to 'not_equal'
    inc si ; to point next element of str from
    dec di   forwated direction.
           ; to point next element of str from
             backword direction
```

```
loop lup        ; end of the loop
mov res 'y'     ; if the string is pallindrome, copy
ret                 ~~res~~ , 'y' to res.

not_equal :
        mov res, 'n'
        ret.

str   db 'Kuet'
len   equ ($-str) ; length of the string
res   db ?
```

Result and discussion:

From this experiment, we learnt about pallind-rome string and also got a cleare idea about string by solving a problem in assembly languag We used almost all the ~~besic~~ basic instruction So this ~~to~~ helped us to practice these instructio

to have cristal clear idea about their implementation. In this program, we used different inputs for checking that whether they are pallindromes or not. We found expected results from this program. So we finally came to the conclusion that our program ~~was~~ performed successfully.

Conclusion:

This experiment was very much necessary for us to practice all the basic instructions and also practice on string type data. We have learnt about pallindrome string and by performing this experiment, we accomplished a good knowledge about assembly language programming.

References:

1. Microprocessor and Interfacing - by D.V. Hall

2. emu 8086 / documentation / index. html.