NAME : MIGUEL CARLOS V. GONZALEZ_______________ **DATE :** _________________________________
SUBJ/SEC : COMPUTER ARCHITECTURE AND ORGANIZATION / BSCOE 4-6 **PROFESSOR : Engr. Rolito L. Mahaguay,** ***MSE-CpE, PCpE***

EXPERIMENT NO. 3
STRING MANIPULATION

OBJECTIVE (S) :
1. To learn how to print strings using service 9 of INT 21h
2. To learn how to process string in the assembly language level
3. To create a program that provides a simple screen output by using the string output service.

REQUIREMENTS:
Personal Computer
TASM
- ❖ Asembler – **TASM.EXE**
- ❖ Loader - **TLINK.EXE**

DISCUSSION:
Strings can't usually fit in a register, strings are then placed in the memory and then pass the address of the string in the memory of two of the registers, the segment address in DS and the offset address in DX.
The string output sends a string of characters to the standard output.
On entry:AH=09h.
DS = segment address of the first character of the string
DX = offset address of the first character of the string.
Service 9 display a string of character starting with the first character (address in DS: DX) output, but not including, the character "$" (ASCII24H)

PROCEDURE:
1. Encode the given program. Assign any filename for experiment # 3. Personalize your filename. Take note that the filename should be a maximum of 8 characters only.

```
                .model small
                .code
                org 100h
start:          jmp main
                x db "RED$"
                y db "BLUE$"

main            proc near
                mov ah, 9
                lea dx, y
                int 21h
                call down
                mov ah, 9
                mov dx, offset x
                int 21h
                int 20h
main            endp

down            proc near
                mov ah, 2
                mov dl,13
                int 21h
                mov dl,10
                int 21h
                ret

down            endp
end             start
```

2. Write down the output of the given program.

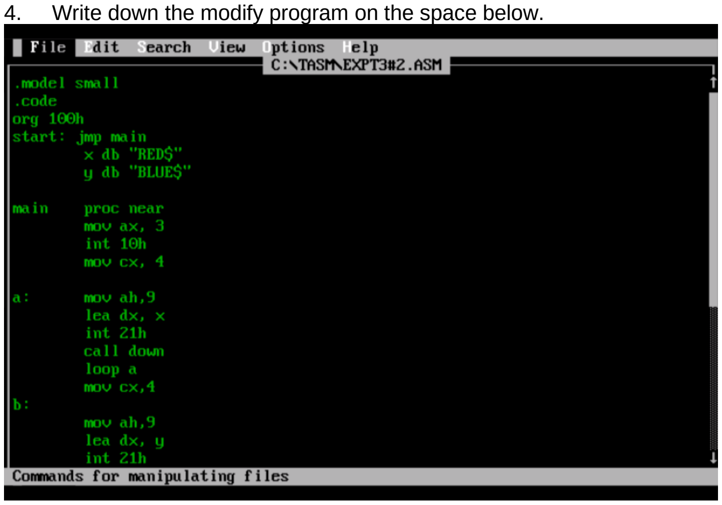
```
RED
BLUE
```

3. Modify eperiment # 3.asm. The output should be:

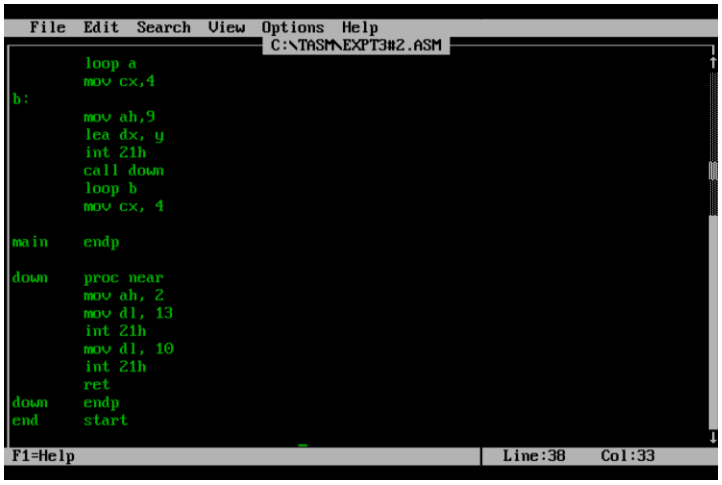
(Note: Use the ***LOOP*** instruction)

RED
RED
RED
RED
BLUE
BLUE
BLUE
BLUE

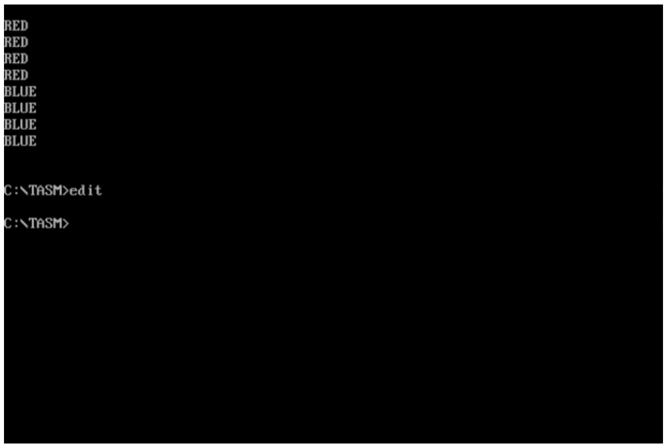
4. Write down the modify program on the space below.


File Edit Search View Options Help
C:\TASM\EXPT3#2.ASM
Commands for manipulating files

```
.model small
.code
org 100h
start: jmp main
        x db "RED$"
        y db "BLUE$"

main    proc near
        mov ax, 3
        int 10h
        mov cx, 4

a:      mov ah,9
        lea dx, x
        int 21h
        call down
        loop a
        mov cx,4
b:
        mov ah,9
        lea dx, y
        int 21h
```



```
        loop a
        mov cx,4
b:
        mov ah,9
        lea dx, y
        int 21h
        call down
        loop b
        mov cx, 4

main    endp

down    proc near
        mov ah, 2
        mov dl, 13
        int 21h
        mov dl, 10
        int 21h
        ret
down    endp
end     start
```



```
RED
RED
RED
RED
BLUE
BLUE
BLUE
BLUE


C:\TASM>edit

C:\TASM>
```

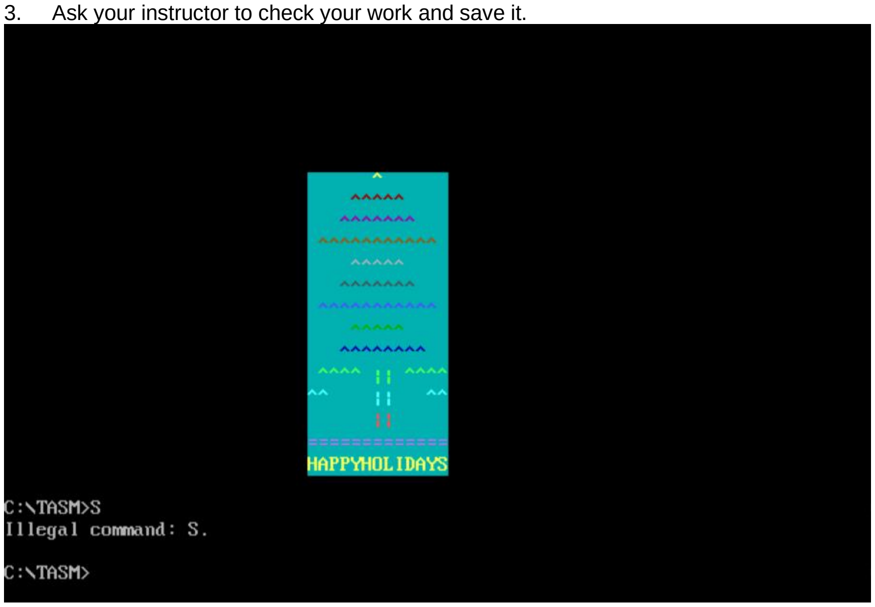
EXERCISES:

1. Write the required program output from your instructor.

2. Encode the program, and then write it in the space below.

    No need to write the source code.

3. Ask your instructor to check your work and save it.



```
      ^
    ^^^^^
   ^^^^^^^
 ^^^^^^^^^^^
    ^^^^^
   ^^^^^^^
 ^^^^^^^^^^^
    ^^^^^
   ^^^^^^^^
 ^^^^ || ^^^^
^^    ||   ^^
      ||
=============
HAPPYHOLIDAYS

C:\TASM>S
Illegal command: S.

C:\TASM>
```

4. What does the NEAR directive mean?
    - Typically associated with x86 assembly programming. Used to specify if a jump or a call function will use a short, near, jump, or call

5. What are the requirements that must be satisfied before INT 21h service 9 prints the string?
    - The string must be identified using double quotes (“ “) when declared.
    - The variable must be declared as type db (data byte)
    - The string must use $ on declaration as to indicate to the program that, special characters are being used.

6. What are the advantages of using procedures?
    - As with other programming languages, procedures are like the functions that we use in today’s programming. It helps us to reuse code and therefore make it more modular.

7. What is the use of RET instruction?

    - This instruction is used to return control from a function to the calling program.

8. Explain how the LOOP instruction works.
    - The loop instruction is used for making loops, it makes use of the CX register to decrement the count; this is similar to a for loop in modern programming languages.

SUMMARY:

Through this experiment, I have learned to use “procedures” which are Assembly’s version of functions. Aside from that, I have also learned how to use loops, and how to implement them to an existing piece of code. Additionally, I learned how to manipulate the appearance of a string, using a combination of hex to modify the colors of the strings and its background.