Hands-On Activity 3.1	
DATA DEFINITION AND TRANSFER	
Course Code: CPE021	**Program:** Computer Engineering
Course Title: Computer Architecture and Organization	**Date Performed:** March 3, 2025
Section: CPE22S2	**Date Submitted:** March 10, 2025
Name: Adia, James Russel E.	**Instructor:** Engr. Maria Rizette H. Sayo

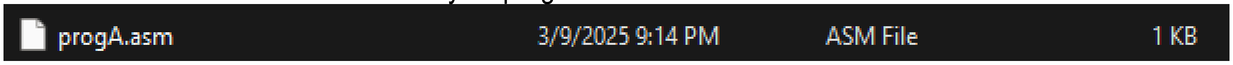
A. Procedure: Output(s) and Observation(s)

Sample Program A.

1. Type the following program using Notepad.

```
dosseg
.model small
.stack
.data
        prompt1 db 13,10,"Enter a character:$"
        prompt2 db 13,10,"The character you entered is:$"
.code
main proc
movax,@data
movds,ax

lea dx,prompt1
mov ah,09h
int 21h

mov ah,01h
int 21h

mov bl,al
lea dx,prompt2
mov ah,09h
int 21h

movdl,bl
mov ah,02h
int 21h
mov ax,4c00h
int 21h

 main endp
end
```

2. Save the file inside the TASM directory as progA.asm.



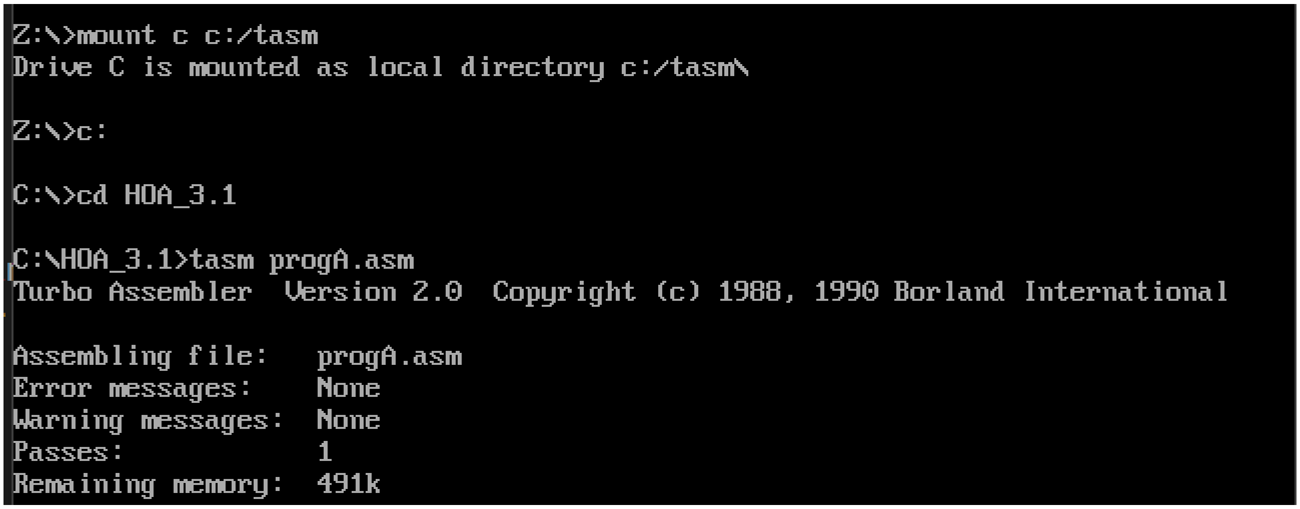
```
progA.asm                3/9/2025 9:14 PM        ASM File              1 KB
```

3. Open the DOS command prompt.
4. Change directory to TASM. Type,
   C:\>**cd TASM <Enter>**
5. Assemble Proga.asm. Type,

C:\tasm>**tasm progA.asm<Enter>**

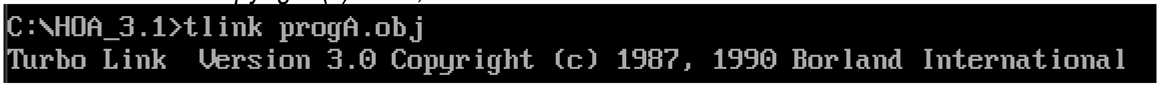
The following message will appear if you assembled your program successfully.

Turbo Assembler Version 2.0 Copyright (c) 1988, 1990 Borland International
Assembling file: proga.ASM
Error messages: None
Warning messages: None
Passes: 1
Remaining memory: 442k



```
Z:\>mount c c:/tasm
Drive C is mounted as local directory c:/tasm\

Z:\>c:

C:\>cd HOA_3.1

C:\HOA_3.1>tasm progA.asm
Turbo Assembler  Version 2.0  Copyright (c) 1988, 1990 Borland International

Assembling file:   progA.asm
Error messages:    None
Warning messages:  None
Passes:            1
Remaining memory:  491k
```

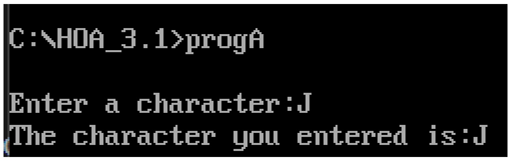
6. Use **tlink** to link all files created from assembling. Type,
   C:\tasm>**tlink ProgA.obj<Enter>**
   Turbo Link Version 3.0 Copyright (c) 1987, 1990 Borland International



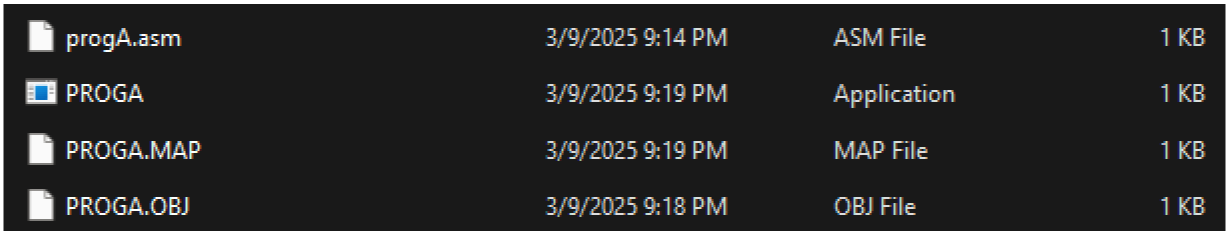
```
C:\HOA_3.1>tlink progA.obj
Turbo Link  Version 3.0 Copyright (c) 1987, 1990 Borland International
```

Execute the program created. Type,
C:\tasm>**ProgA<Enter>**

7. Analyze and record the output in Table 3.2.



```
C:\HOA_3.1>progA

Enter a character:J
The character you entered is:J
```



progA.asm	3/9/2025 9:14 PM	ASM File	1 KB
PROGA	3/9/2025 9:19 PM	Application	1 KB
PROGA.MAP	3/9/2025 9:19 PM	MAP File	1 KB
PROGA.OBJ	3/9/2025 9:18 PM	OBJ File	1 KB

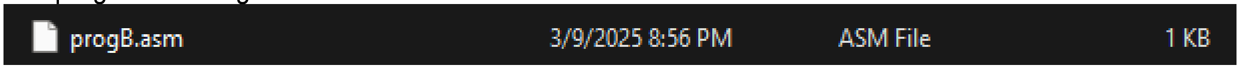
Sample Program B.

1. Type the following program using Notepad.

```
.model small
.stack 100h
.data
byte1 db 1
byte2 db 0
```

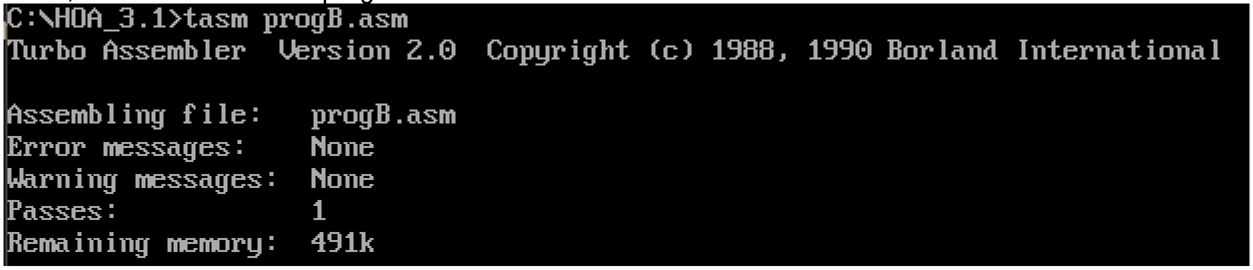
```
word1 dw 1234h
word2  dw 0
string db "Stressed!", 0dh, 0ah,"$"
;---- this is a comment
 .code
 MAIN PROC
Mov ax, @data
Mov ds, ax

Mov dx, offset string
Mov ah, 9
Int 21h

Movbx, offset string
Mov al, [bx]
Mov ah, [bx+1]
Mov [bx], ah
Mov [bx+1],al
Mov ah,9
Mov dx, offset string
Int 21h

Mov ax, 4c00h
Int 21h

Main endp
End main
```

2. Save the program as ProgB.asm.

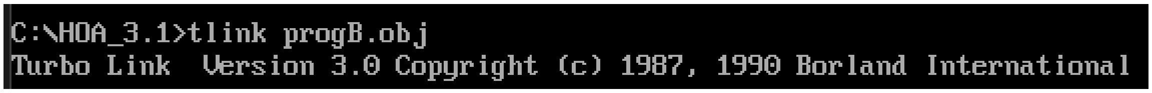


```
progB.asm          3/9/2025 8:56 PM          ASM File          1 KB
```

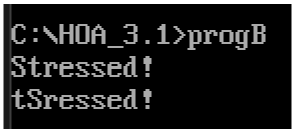
3. Assemble, link and execute the program.



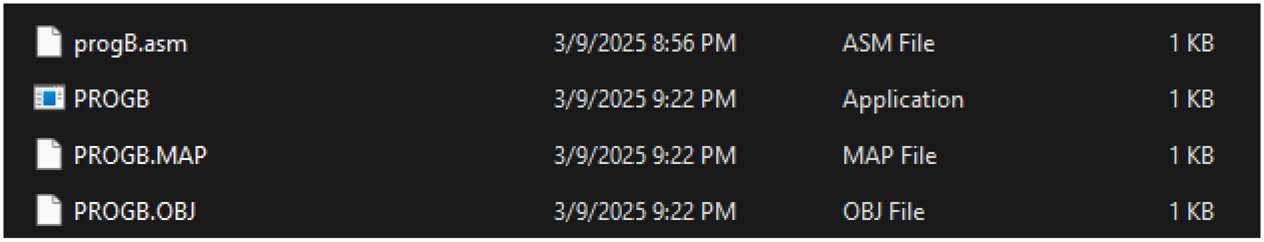
```
C:\HOA_3.1>tasm progB.asm
Turbo Assembler  Version 2.0  Copyright (c) 1988, 1990 Borland International

Assembling file:   progB.asm
Error messages:    None
Warning messages:  None
Passes:            1
Remaining memory:  491k
```



```
C:\HOA_3.1>tlink progB.obj
Turbo Link  Version 3.0 Copyright (c) 1987, 1990 Borland International
```

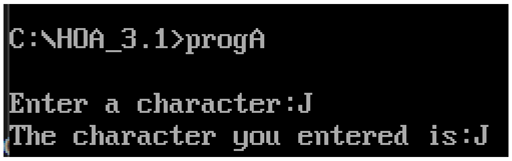
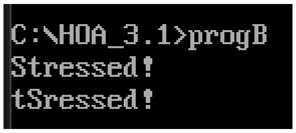


```
C:\HOA_3.1>progB
Stressed!
tSressed!
```



progB.asm	3/9/2025 8:56 PM	ASM File	1 KB
PROGB	3/9/2025 9:22 PM	Application	1 KB
PROGB.MAP	3/9/2025 9:22 PM	MAP File	1 KB
PROGB.OBJ	3/9/2025 9:22 PM	OBJ File	1 KB

4. Analyze and record the output in Table 3.3

Table 3.2- Output of Sample Program A	*Table 3.3- Output of Sample Program B*
 C:\HOA_3.1>progA Enter a character:J The character you entered is:J - I observed that this program demonstrates basic DOS interrupt handling for displaying strings and reading characters from the keyboard	 C:\HOA_3.1>progB Stressed! tSressed! - Just like the Output of Sample Program A. I also observed that this program demonstrates basic DOS interrupt handling for displaying strings. But this time, instead if reading characters from the keyboard based from user input, it specifically shows how to swap the first two characters of a string and then it displays the result.

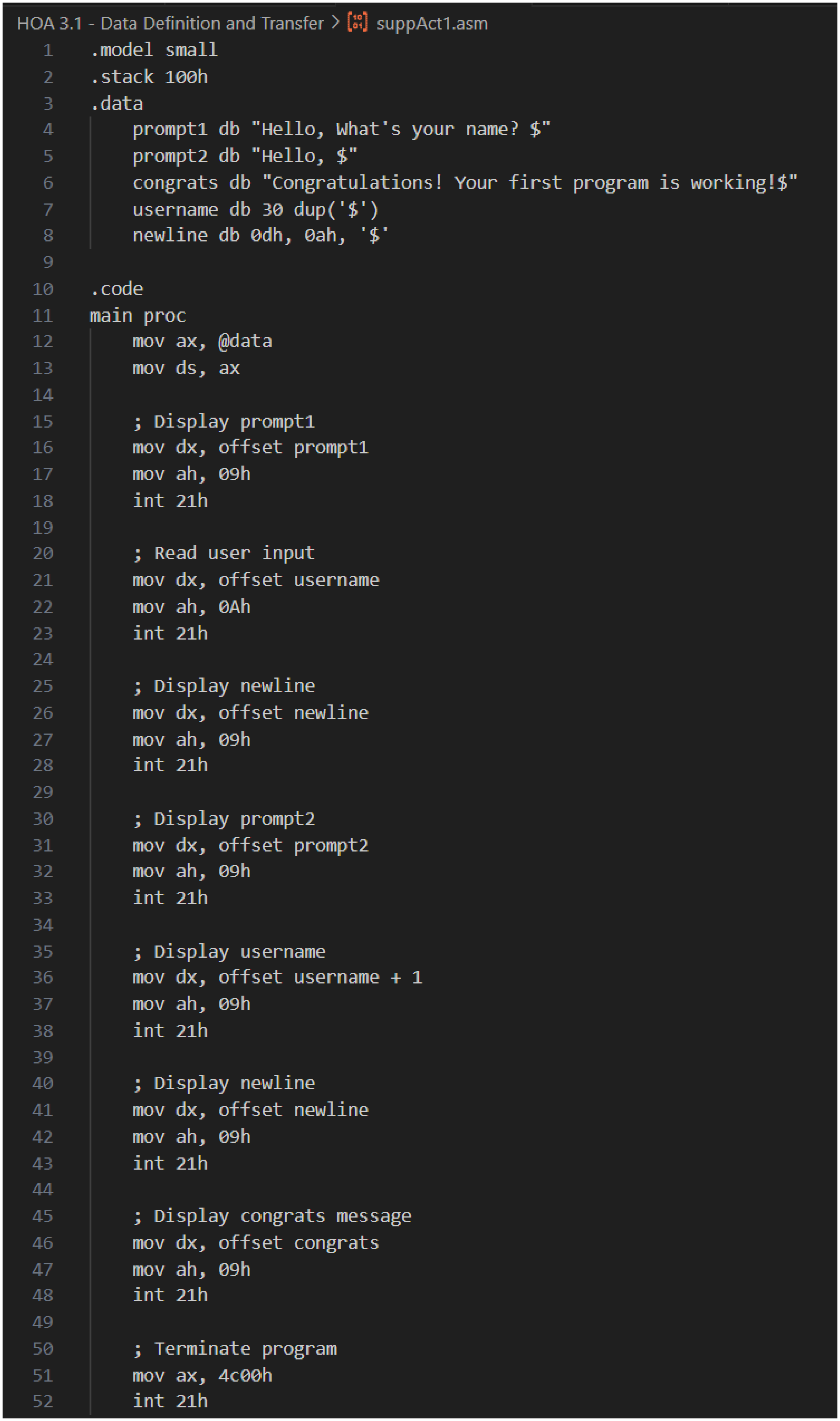
B. Supplementary Activity: Output(s) and Observation(s)

1. How many bytes are allocated for each of the following data definitions?
   a. BYTE 20 DUP(0)= 20 bytes
   b. BYTE 20 DUP (?)= 20 bytes
   c. BYTE 4 DUP("East")= 16 bytes
   d. WORD3 WORD ?= 6 bytes
   e. Array WORD 5 DUP(?)= 10 bytes

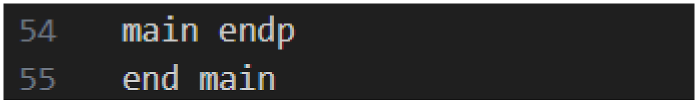
2. Create a program that prompts and reads a user's name USERNAME (of maximum length 30 characters). The program should display a message of the form:

OUTPUT:
Hello, What's your name? JM
Hello, JM
Congratulations! Your first program is working!

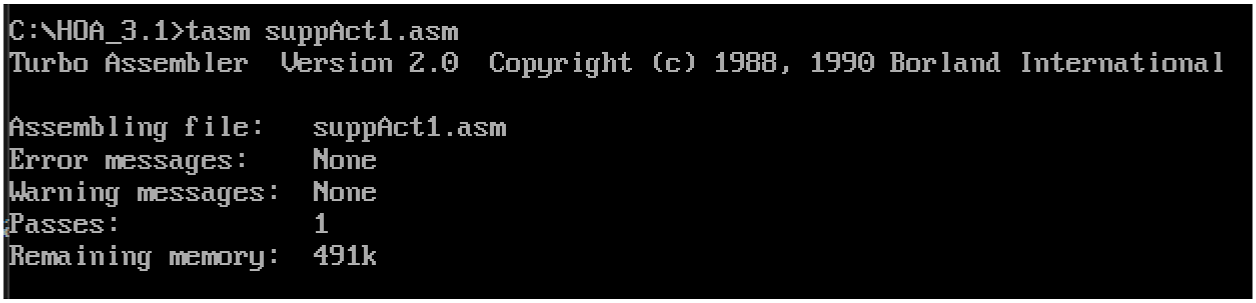
Program Screenshot (.asm file opened with vscode for better readability):



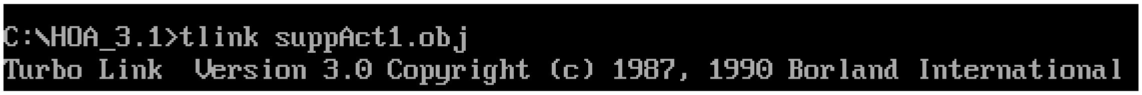
```
.model small
.stack 100h
.data
    prompt1 db "Hello, What's your name? $"
    prompt2 db "Hello, $"
    congrats db "Congratulations! Your first program is working!$"
    username db 30 dup('$')
    newline db 0dh, 0ah, '$'

.code
main proc
    mov ax, @data
    mov ds, ax

    ; Display prompt1
    mov dx, offset prompt1
    mov ah, 09h
    int 21h

    ; Read user input
    mov dx, offset username
    mov ah, 0Ah
    int 21h

    ; Display newline
    mov dx, offset newline
    mov ah, 09h
    int 21h

    ; Display prompt2
    mov dx, offset prompt2
    mov ah, 09h
    int 21h

    ; Display username
    mov dx, offset username + 1
    mov ah, 09h
    int 21h

    ; Display newline
    mov dx, offset newline
    mov ah, 09h
    int 21h

    ; Display congrats message
    mov dx, offset congrats
    mov ah, 09h
    int 21h

    ; Terminate program
    mov ax, 4c00h
    int 21h
```



```
main endp
end main
```

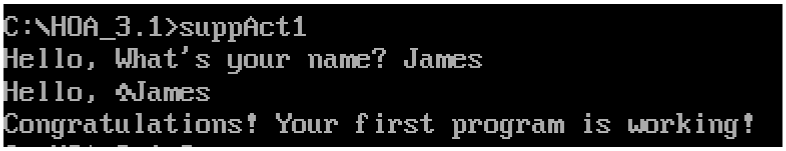


```
C:\HOA_3.1>tasm suppAct1.asm
Turbo Assembler  Version 2.0  Copyright (c) 1988, 1990 Borland International

Assembling file:   suppAct1.asm
Error messages:    None
Warning messages:  None
Passes:            1
Remaining memory:  491k
```



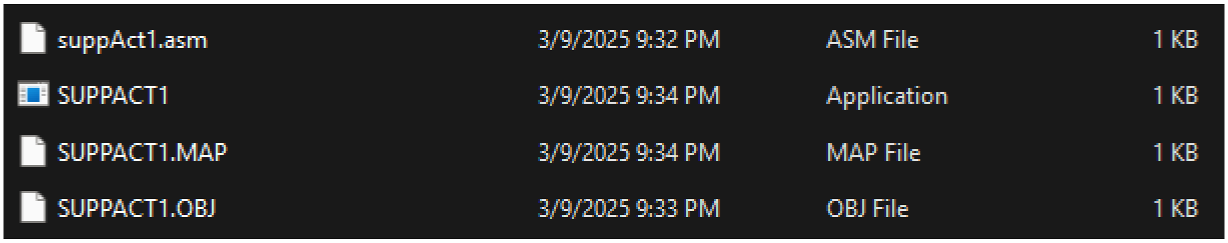
```
C:\HOA_3.1>tlink suppAct1.obj
Turbo Link  Version 3.0 Copyright (c) 1987, 1990 Borland International
```

suppAct1.exe Program Output:



```
C:\HOA_3.1>suppAct1
Hello, What's your name? James
Hello, ♣James
Congratulations! Your first program is working!
```

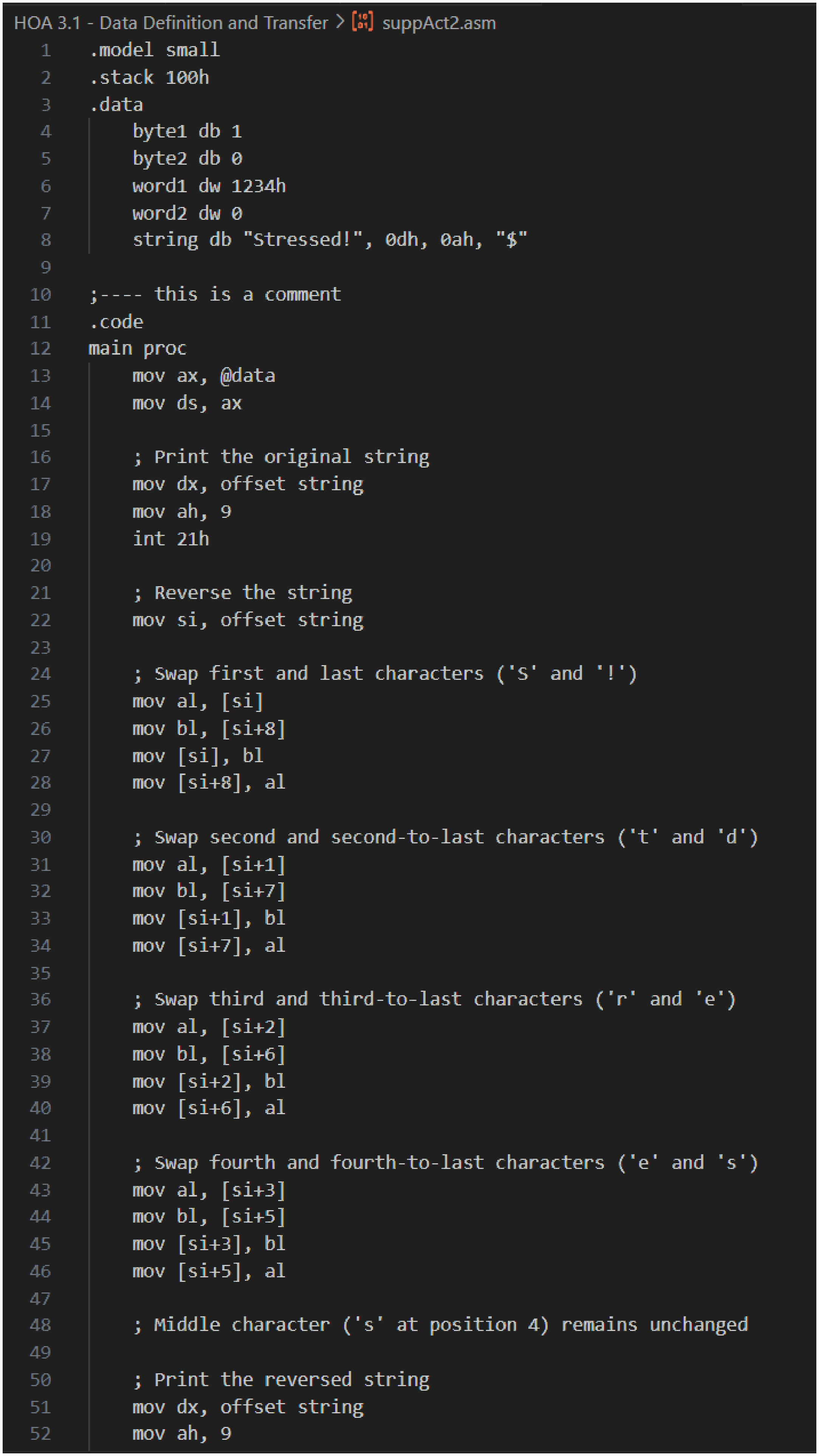
Files Created:



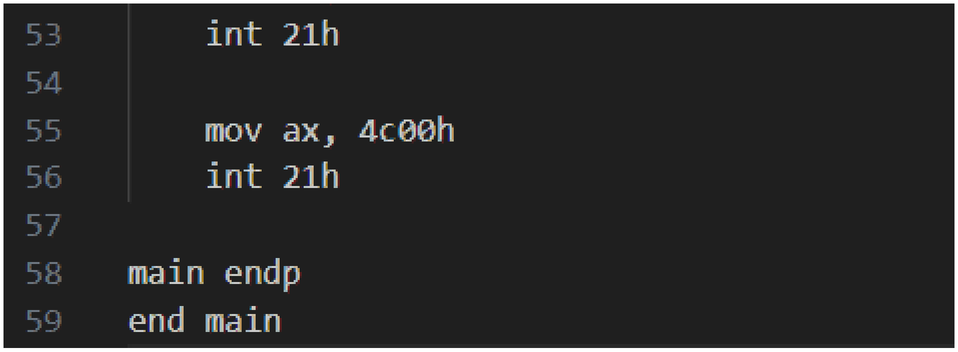
suppAct1.asm	3/9/2025 9:32 PM	ASM File	1 KB
SUPPACT1	3/9/2025 9:34 PM	Application	1 KB
SUPPACT1.MAP	3/9/2025 9:34 PM	MAP File	1 KB
SUPPACT1.OBJ	3/9/2025 9:33 PM	OBJ File	1 KB

3. Modify ProgB such that the second string is printed "!dessertS" ("Stressed!" backwards).

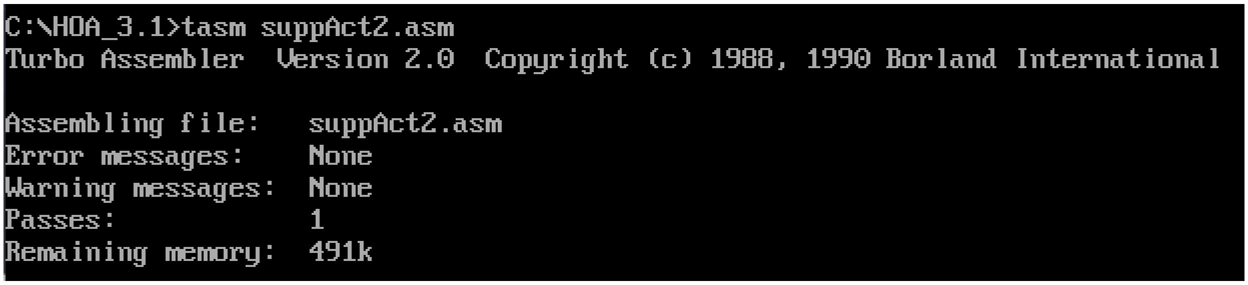
Program Screenshot (.asm file opened with vscode for better readability):



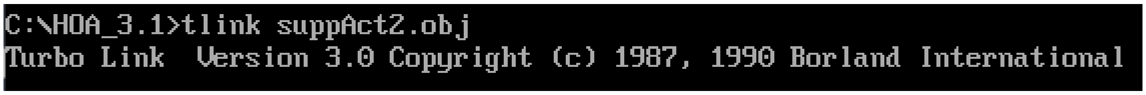
```
.model small
.stack 100h
.data
    byte1 db 1
    byte2 db 0
    word1 dw 1234h
    word2 dw 0
    string db "Stressed!", 0dh, 0ah, "$"

;---- this is a comment
.code
main proc
    mov ax, @data
    mov ds, ax

    ; Print the original string
    mov dx, offset string
    mov ah, 9
    int 21h

    ; Reverse the string
    mov si, offset string

    ; Swap first and last characters ('S' and '!')
    mov al, [si]
    mov bl, [si+8]
    mov [si], bl
    mov [si+8], al

    ; Swap second and second-to-last characters ('t' and 'd')
    mov al, [si+1]
    mov bl, [si+7]
    mov [si+1], bl
    mov [si+7], al

    ; Swap third and third-to-last characters ('r' and 'e')
    mov al, [si+2]
    mov bl, [si+6]
    mov [si+2], bl
    mov [si+6], al

    ; Swap fourth and fourth-to-last characters ('e' and 's')
    mov al, [si+3]
    mov bl, [si+5]
    mov [si+3], bl
    mov [si+5], al

    ; Middle character ('s' at position 4) remains unchanged

    ; Print the reversed string
    mov dx, offset string
    mov ah, 9
```



```
    int 21h

    mov ax, 4c00h
    int 21h

main endp
end main
```

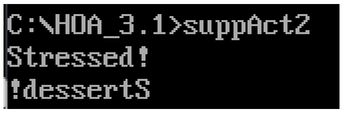


```
C:\HOA_3.1>tasm suppAct2.asm
Turbo Assembler  Version 2.0  Copyright (c) 1988, 1990 Borland International

Assembling file:   suppAct2.asm
Error messages:    None
Warning messages:  None
Passes:            1
Remaining memory:  491k
```



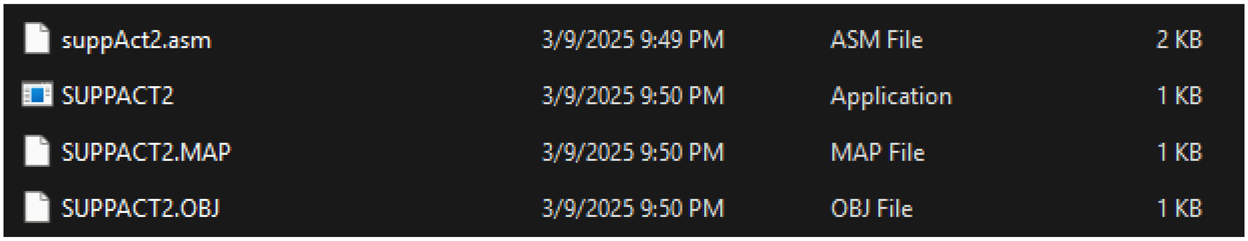
```
C:\HOA_3.1>tlink suppAct2.obj
Turbo Link  Version 3.0 Copyright (c) 1987, 1990 Borland International
```

suppAct2.exe Program Output:



```
C:\HOA_3.1>suppAct2
Stressed!
!dessertS
```

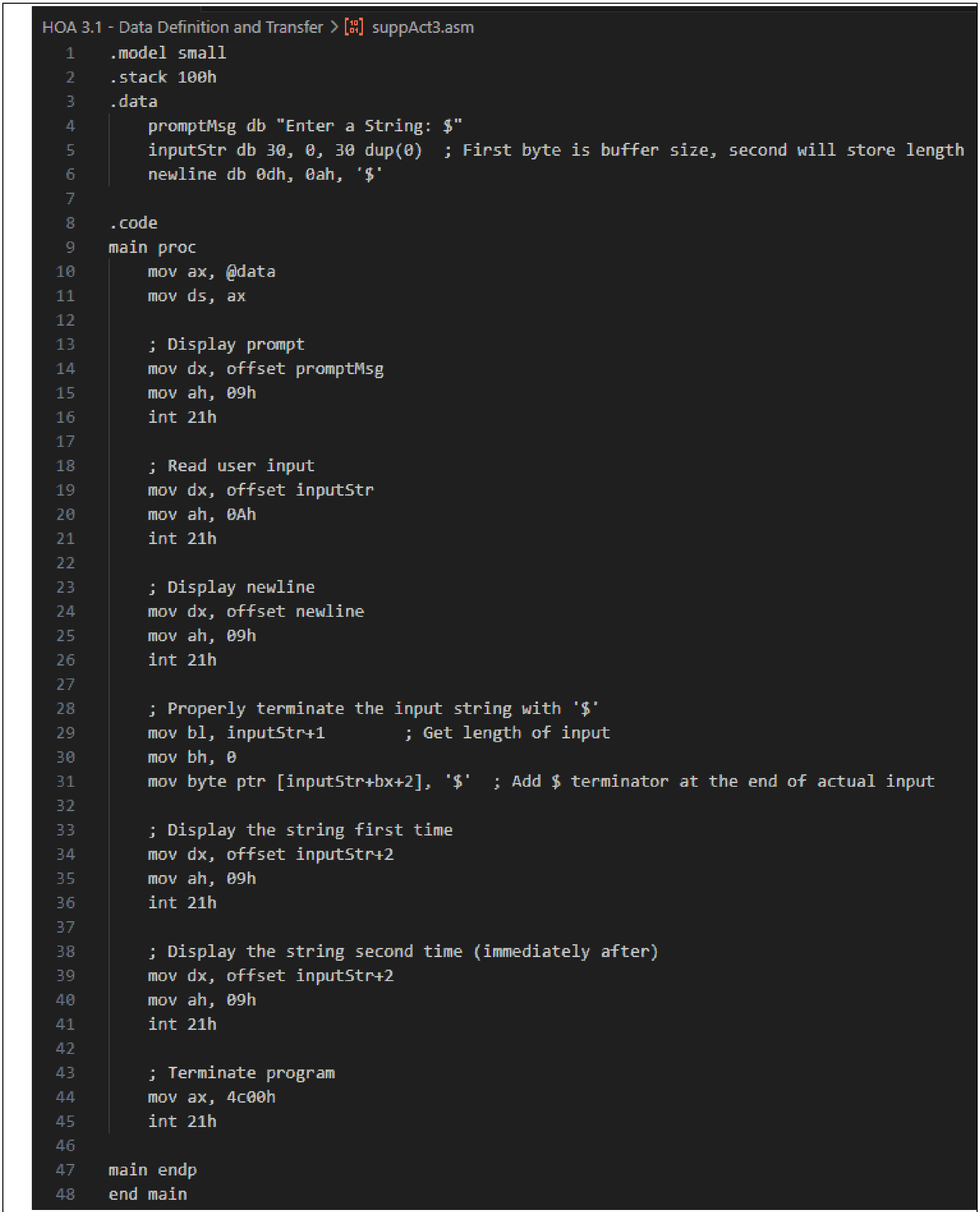
Files Created:



suppAct2.asm	3/9/2025 9:49 PM	ASM File	2 KB
SUPPACT2	3/9/2025 9:50 PM	Application	1 KB
SUPPACT2.MAP	3/9/2025 9:50 PM	MAP File	1 KB
SUPPACT2.OBJ	3/9/2025 9:50 PM	OBJ File	1 KB

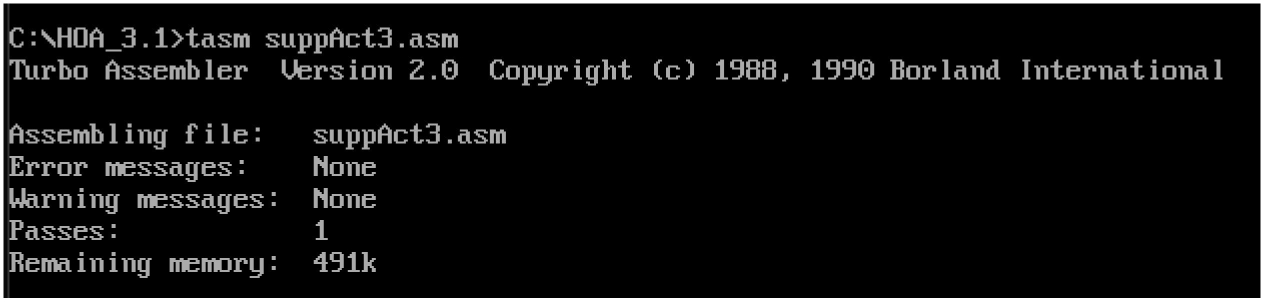
4. Create a program that will display a given string two times.
   SAMPLE OUTPUT:
   Enter a String: **love**
   lovelove

Program Screenshot (.asm file opened with vscode for better readability):

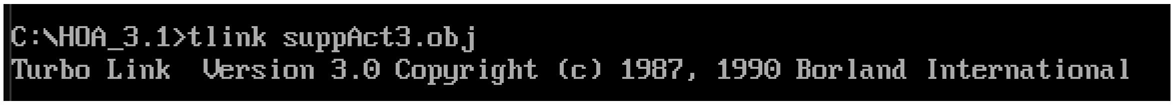


HOA 3.1 - Data Definition and Transfer > suppAct3.asm

```
.model small
.stack 100h
.data
    promptMsg db "Enter a String: $"
    inputStr db 30, 0, 30 dup(0)  ; First byte is buffer size, second will store length
    newline db 0dh, 0ah, '$'

.code
main proc
    mov ax, @data
    mov ds, ax

    ; Display prompt
    mov dx, offset promptMsg
    mov ah, 09h
    int 21h

    ; Read user input
    mov dx, offset inputStr
    mov ah, 0Ah
    int 21h

    ; Display newline
    mov dx, offset newline
    mov ah, 09h
    int 21h

    ; Properly terminate the input string with '$'
    mov bl, inputStr+1        ; Get length of input
    mov bh, 0
    mov byte ptr [inputStr+bx+2], '$'  ; Add $ terminator at the end of actual input

    ; Display the string first time
    mov dx, offset inputStr+2
    mov ah, 09h
    int 21h

    ; Display the string second time (immediately after)
    mov dx, offset inputStr+2
    mov ah, 09h
    int 21h

    ; Terminate program
    mov ax, 4c00h
    int 21h

main endp
end main
```

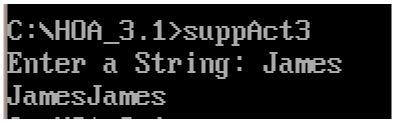


```
C:\HOA_3.1>tasm suppAct3.asm
Turbo Assembler  Version 2.0  Copyright (c) 1988, 1990 Borland International

Assembling file:   suppAct3.asm
Error messages:    None
Warning messages:  None
Passes:            1
Remaining memory:  491k
```



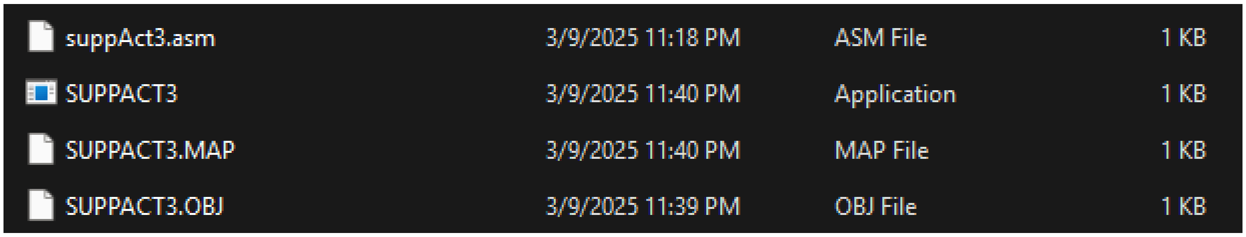
```
C:\HOA_3.1>tlink suppAct3.obj
Turbo Link  Version 3.0 Copyright (c) 1987, 1990 Borland International
```

SuppAct3.exe Program Output:



```
C:\HOA_3.1>suppAct3
Enter a String: James
JamesJames
```

Files Created:



suppAct3.asm	3/9/2025 11:18 PM	ASM File	1 KB
SUPPACT3	3/9/2025 11:40 PM	Application	1 KB
SUPPACT3.MAP	3/9/2025 11:40 PM	MAP File	1 KB
SUPPACT3.OBJ	3/9/2025 11:39 PM	OBJ File	1 KB

C. Conclusion & Lessons Learned

After doing the Hands-On activity, I was able to create assembly programs that demonstrates the intended learning outcomes by showcasing techniques for defining data and manipulating character strings. The programs created includes examples of declaring groups of characters while utilizing DOS interrupt 21h with service 09h to display strings. Through the procedures and supplementary activities, I gained practical experience in writing assembly code that allows for user input and outputs the user input.