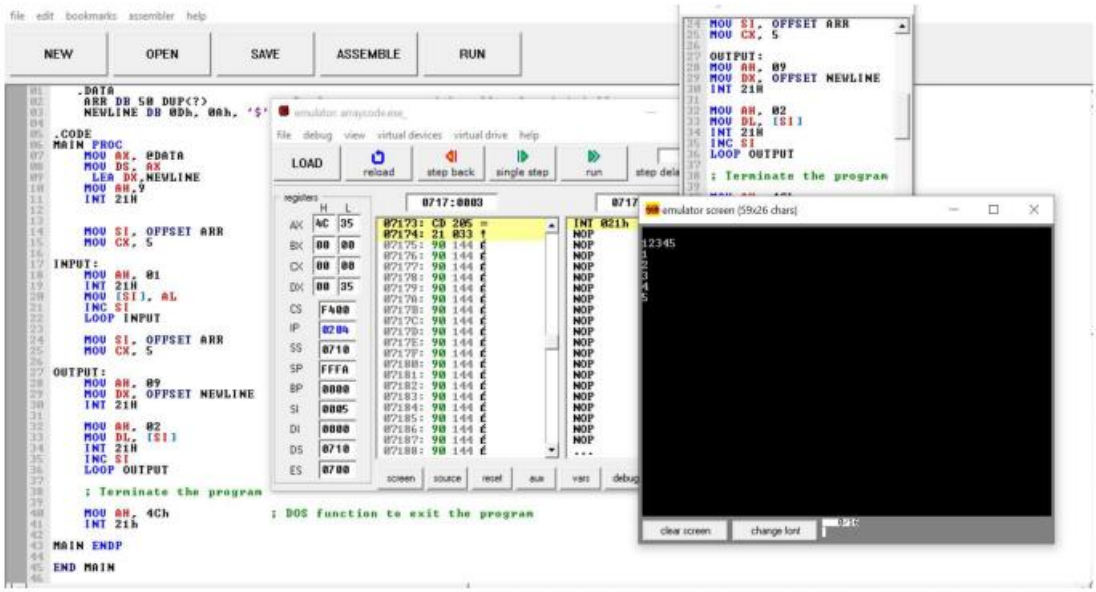
Code:

```
.DATA
 ARR DB 50 DUP(?) ; Declare an array with null value initially
 NEWLINE DB 0Dh, 0Ah, '$' ; Define a new line string
.CODE
MAIN PROC
 MOV AX, @DATA ; Load the data segment address into AX
 MOV DS, AX
 LEA DX, NEWLINE
 MOV AH,9
 INT 21H
 ; Set DS to the data segment
 MOV SI, OFFSET ARR ; Load the address of the array
 MOV CX, 5 ; Set the counter to 5 for 5 characters
INPUT:
 MOV AH, 01 ; DOS function to read a character
 INT 21H ; Invoke interrupt
 MOV [SI], AL ; Store the character in the array
 INC SI ; Increment the index
 LOOP INPUT ; Repeat until CX = 0
6
 MOV SI, OFFSET ARR ; Load the address of the array again
 MOV CX, 5 ; Reset the counter
OUTPUT:
 MOV AH, 09 ; DOS function to print a string
 MOV DX, OFFSET NEWLINE ; Load the address of the new line
```

```
string
 INT 21H ; Invoke interrupt
 MOV AH, 02 ; DOS function to print a character
 MOV DL, [SI] ; Load the character from the array
 INT 21H ; Invoke interrupt
 INC SI ; Increment the index
 LOOP OUTPUT ; Repeat until CX = 0
 ; Terminate the program

 MOV AH, 4Ch ; DOS function to exit the program
 INT 21h
MAIN ENDP
END MAIN
```

Results:


file edit bookmarks assembler help
NEW
OPEN
SAVE
ASSEMBLE
RUN
.DATA
ARR DB 50 DUP(?)
NEWLINE DB 0Dh, 0Ah, '$'
.CODE
MAIN PROC
MOV AX, @DATA
MOV DS, AX
LEA DX,NEWLINE
MOV AH,9
INT 21H
MOV SI, OFFSET ARR
MOV CX, 5
INPUT:
MOV AH, 01
INT 21H
MOV [SI], AL
INC SI
LOOP INPUT
MOV SI, OFFSET ARR
MOV CX, 5
OUTPUT:
MOV AH, 09
MOV DX, OFFSET NEWLINE
INT 21H
MOV AH, 02
MOV DL, [SI]
INT 21H
INC SI
LOOP OUTPUT
; Terminate the program
MOV AH, 4Ch ; DOS function to exit the program
INT 21h
MAIN ENDP
END MAIN
emulator screen (59x26 chars)
12345
1
2
3
4
5
clear screen
change font