```
                .ORIG x0000

START:          LEA     R0,MYMSG        ; Load the Message Effective Address
                JSR     PUTSMSG         ; Call the Puts Message subroutine (PUTS)
AGAIN:          JSR     GETCHAR         ; Call the Get Char subroutine (GETC)
                JSR     PUTCHAR         ; Call the Put Char suboutine (OUT)
END:            BR      AGAIN           ; Wait for another character

GETCHAR:        ?                       ; Read the Keyboard Status Register
                                          (KBSR) to check if there is a new char
                                          available (x8000)
                ?                       ; If KBSR != x8000, jump to GetChar
                ?                       ; Read the Keyboard Data Register (KBDR)
                                          to take the incoming character
                ?                       ; Subroutine return

PUTCHAR:        ?                       ; Store R0 into memory to keep a copy of
                                          the incoming character
PUTCHAR2:       ?                       ; Read the Display Status Register (DSR)
                                          to check if a character can be
                                          transmitted (x8000)
                ?                       ; If (DSR != x8000), jump to PutChar2
                ?                       ; Restore the character taken from the
                                          Keyboard to be sent to the display
                ?                       ; Write the Display Data Register (DDR)
                                          with the character taken from the
                                          Keyboard
                ?                       ; Subroutine return
PCR0:           .FILL   0

PUTSMSG:        ?                       ; Store R0 into memory to keep a copy of
                                          the next char address
                ?                       ; Load the char to be sent
                ?                       ; Return if the char is NULL
                ?                       ; Store R7 because is needed by RET
                                          instruction
                ?                       ; Send the char in R0
                ?                       ; Restore R7
                ?                       ; Restore the address of the char sent
                ?                       ; Compute the address of the next char
                ?                       ; Send the next char
PUTSMSGE:       ?                       ; Subroutine return
PMR0            .FILL   0
PMR7            .FILL   0

KBSR:           .FILL   xFE00           ; Keyboard Status Register Address
KBDR:           .FILL   xFE02           ; Keyboard Data Register Address
DSR:            .FILL   xFE04           ; Display Status Register Address
DDR:            .FILL   xFE06           ; Display Data Register Address
MYMSG:          .STRINGZ "\nHello, welcome to the ConsoleEchoing program
                          test.\n\nPlease type any char you want to echo: "
                .END
```