```
.ORIG X3000

                        JSR LENGTH  ;FINDS LENGTH OF STRING TO BE GUESSED.
MAIN                    JSR OUTPUT
                        LEA R0, CHARASK
                        JSR PUT
                        JSR GET
                        JSR NEWLINE
                        JSR CHECK
                        JSR COMPSTR
                        BR MAIN

OUTPUT                  LEA R0, PROMPT
                        LEA R6, GUESSEDWORD
                        STI R7, JSRSTORAGE ;STORES R7 SO IT CAN BE LOADED LATER.
                        JSR PUT
                        AND R0, R0, #0
                        ADD R0, R6, #0
                        JSR PUT
                        JSR NEWLINE
                        LDI R7, JSRSTORAGE
                        RET

CHECK                   AND R2, R2, #0 ;COUNTING VARIABLE
                        LEA R6, WORDTOGUESS
                        LDI R5, CORRECTSTORAGE
CHECKLOOP               LDR R3, R6, #0
                        BRZ ENDCHECK
                        NOT R3, R3
                        ADD R3, R3, #1
                        ADD R3, R3, R0
                        BRNP NOMATCH
                        LEA R4, GUESSEDWORD ;IF IT DOES MATCH
                        ADD R4, R2, R4      ;ADDS INDEX OF CHAR TO REPLACE.
                        STR R0,R4,#0
                        STI R0,GUESSEDWORD  ;REPLACES WITH CORRECT CHAR.
                        ADD R5, R5, #1  ;ADDS TO CORRECT GUESS COUNT.
                        STI R5,CORRECTSTORAGE
NOMATCH                 ADD R2, R2, #1 ;INCREMENT INDEX COUNTER
                        ADD R6, R6, #1
                        BRNZP CHECKLOOP
ENDCHECK                STI R5,CORRECTSTORAGE
                        RET

COMPSTR                 AND R6, R6, #0
                        LEA R0, WORDTOGUESS
                        LEA R1, GUESSEDWORD
COMPLOOP                LDR R2, R0, #0
                        LDR R3, R1, #0
                        BRZ WIN         ;IF THIS BRANCHES, IT MEANS
                        NOT R3, R3      ;THAT NO CHARACTER DIDNT’ MATCH
                        ADD R3, R3, #1  ;WORDTOGUESS AND GUESSEDWORD MUST BE
                        ADD R3, R3, R2  ;SAME LENGTH THOUGH.
                        BRNP EXITCOMP   ;char did not match
                        ADD R0, R0, #1
                        ADD R1, R1, #1
                        ADD R6, R6, #0
                        BR COMPLOOP
EXITCOMP                RET

;R0 HOLDS ADDRESS OF STRING TO BE PUT
PUT                     LDI R1, DSR
                        BRZP PUT
                        LDR R2, R0, #0  ;LOAD STRING ELEMENT INTO R2
                        BRZ ENDPUT      ;IF STRING IS DONE, RETURN
                        STI R2, DDR
                        ADD R0, R0, #1
                        BR PUT
ENDPUT                  RET


;R0 WILL HOLD VALUE READ FROM KEYBOARD
GET                     LDI R1, KBSR
                        BRZP GET
                        LDI R0, KBDR
ECHO                    LDI R1, DSR
                        BRZP ECHO
                        STI R0, DDR
                        RET

;PRINTS NEW LINE
NEWLINE                 LDI R5, DSR
                        BRZP NEWLINE
                        LD R6, NEWCHAR
                        STI R6, DDR
                        RET

;CALCULATES THE LENGTH OF THE WORD NEEDED TO GUESS.
LENGTH                  AND R3, R3, #0
                        LEA R0, WORDTOGUESS
LLOOP                   LDR R1, R0, #0
                        BRZ ENDLENGTH
                        ADD R3, R3, #1 ;INCREMENT STRING LENGTH COUNTER
                        ADD R0, R0, #1 ;ITERATE THRU STRING.
                        BR LLOOP
ENDLENGTH               STI R3, GUESSEDSTORAGE
                        RET

WIN                     JSR OUTPUT
                        JSR NEWLINE
                        LEA R0, WINSTR
                        JSR PUT
                        HALT



;program data

WordToGuess .stringz "queue"
GuessedWord .stringz "_____"
CorrectStorage .fill x4000
GuessedStorage .fill x4001
JSRStorage .fill x4002
Prompt .stringz "Word = "
CharAsk .stringz "Enter >"
WinStr  .stringz "You win!"

DSR     .fill xFE04
DDR     .fill xFE06
KBSR    .fill xFE00
KBDR    .fill xFE02
NewChar .fill x0A
.end
```