Problem Statement: 8051 project that converts ASCII alphabet to Morse code on port 0 of the 8051.

Algorithm: The look table will be used to compare and find the equivalent Morse code value. First, the equivalent value from ASCII is found by subtracting 41H from the value given and checking the value.
A dot (.) is represented as 01
A dash (-) is represented as 11
The EXCHANGINGDPTR subroutine manages the exchange of DPTR values with temporary registers. This allows the program to switch between accessing the LUT and the input string without losing track of their respective addresses.

The letter 'A' is represented in Morse code as. -, which translates to the binary sequence 0111. This binary sequence is stored in the LUT at the index corresponding to 'A'.

Code:

```
ORG 0H
START:
   MOV DPTR, #LUT          ; Set DPTR to the lookup table (LUT) address
         ACALL EXCHANGINGDPTR     ; Exchange DPTR with temporary registers
   MOV DPTR, #DATATOCONVERT ; Set DPTR to the address of the data to convert (ASCII characters)


NEXT_CHAR:
```

```
        MOV A, #00H             ; Clear register A
    MOVC A, @A+DPTR         ; Load the current character from DPTR (ASCII value)
    JZ END_PROGRAM          ; If the character is 0 end the program
    ACALL ASCII_TO_MORSE    ; Call the subroutine to convert ASCII to Morse
    MOV P0, A               ; Output the result to Port 0 (P0)
    INC DPTR                ; Increment DPTR to point to the next character
    SJMP NEXT_CHAR          ; Repeat for the next character


END_PROGRAM:
    SJMP $                  ; Infinite loop to end the program


ASCII_TO_MORSE:

        MOV A, #00H             ; Clear register A
    MOVC A, @A+DPTR         ; Load the ASCII value from DPTR
        clr c
    SUBB A, #41H ; Subtract 41H (ASCII value of 'A') to get index for Morse LUT
    CJNE A, #25, OUT        ; If the result is not within range (A-Z), jump to OUT
    MOV A, #00H             ; If the character is not A-Z, clear A

OUT:
        ACALL EXCHANGINGDPTR    ; Swap DPTR with temporary registers
    MOVC A, @A+DPTR         ; Load the Morse code from the LUT using the adjusted index
        ACALL EXCHANGINGDPTR    ; Swap DPTR back to the original value
    RET                     ; Return from the subroutine


LUT:
    DB 0x70     ;A 01110000
    DB 0xD5
    DB 0xDD
    DB 0xD4
    DB 0x04
    DB 0x5D
    DB 0xF4
    DB 0x55    ; H 01010101
    DB 0x05 ;I 010100000
    DB 0x7F             ;J
    DB 0xDC  ;K
    DB 0x75    ;L
    DB 0xF0      ;M
    DB 0xD0       ;N 1101
    DB 0xFC             ; O 11111100
    DB 0x5D
    DB 0xF7
    DB 0x74    ;R 01110100
```

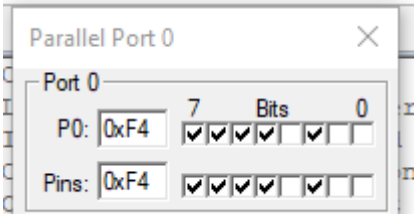
```
    DB 0x54   ;S 01010100
    DB 0x0C ;T 1 1100-C
    DB 0x5C   ;u 001
    DB 0x57   ;v 0001
    DB 0x7C    ;w
    DB 0xD7   ;x
    DB 0xDF   ;y 1011
    DB 0xF5
DATATOCONVERT:
        DB "GURKIRAT", 0

EXCHANGINGDPTR:
        ACALL CLEARA
        MOV A, DPL
    MOV R2, A
    MOV A, DPH
    MOV R3, A
    MOV A, R0
    MOV DPL, A
    MOV A, R2
    MOV R0, A
    MOV A, R1
    MOV DPH, A
    MOV A, R3
    MOV R1, A
        ACALL GETA
    RET
CLEARA:
        mov R5,a
        mov a,#00h
        ret
GETA:
        mov a,R5
        ret
        END
```

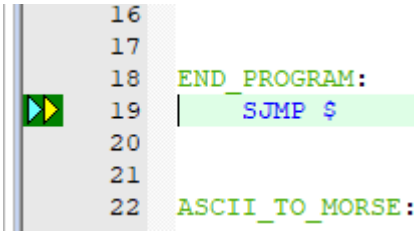
Input and output:

```
DATATOCONVERT:
    DB "GURKIRAT", 0
```

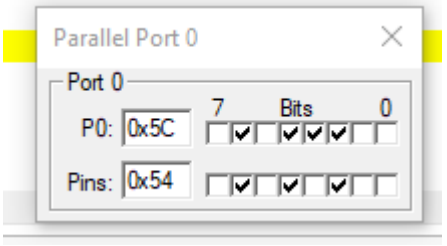
Input



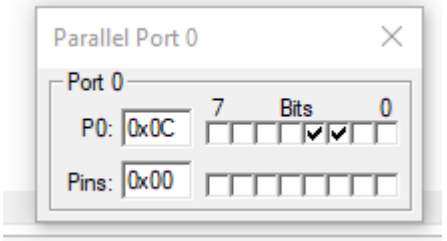
G Morse Code output



Infinite loop after end of input text



U morse code-dot dot dash



T morse code - dash