Este código é uma implementação para exibir números no formato LCD no terminal.

O formato LCD utiliza segmentos horizontais e verticais para formar os dígitos de 0 a 9,

como em uma calculadora.

Estrutura do Código:

1. Dicionário de Segmentos:

   - Define quais segmentos (top, top-left, etc.) são ativados para cada dígito de 0 a 9.

2. Função process_input:

   - Recebe o tamanho (s) e o número (n) como entrada.

   - Calcula a largura total necessária para a tela.

   - Cria uma matriz (screen) que representa a tela como uma grade 2D.

   - Para cada dígito em n, desenha os segmentos ativados na matriz.

3. Função show_number:

   - Recebe a matriz screen e imprime linha por linha no terminal.

4. Loop Principal:

   - Solicita ao usuário o tamanho (s) e o número (n).

   - O programa continua até que s = 0 e n = "0" sejam inseridos.

   - Para cada entrada válida, chama a função process_input.

Observação:

O tamanho s define a escala do número. Quanto maior o valor de s, maior será o tamanho do número na saída.

Nota adicional:

A ideia de implementar a tela como uma matriz/tabela foi inspirada ao observar a implementação em C++ disponível no seguinte repositório:

https://github.com/Sharknevercries/Online-Judge/blob/master/UVA/Volume%20VII/706%20LCD%20Display.cpp

DEFINE segments COMO um dicionário:

'0' -> [top, top-left, top-right, bottom-left, bottom-right, bottom]

'1' -> [top-right, bottom-right]

'2' -> [top, top-right, middle, bottom-left, bottom]

'3' -> [top, top-right, middle, bottom-right, bottom]

'4' -> [top-left, top-right, middle, bottom-right]

'5' -> [top, top-left, middle, bottom-right, bottom]

'6' -> [top, top-left, middle, bottom-left, bottom-right, bottom]

'7' -> [top, top-right, bottom-right]

'8' -> [all segments]

'9' -> [top, top-left, top-right, middle, bottom-right, bottom]

FUNÇÃO show_number(screen):

PARA cada linha EM screen:

IMPRIMA a linha como uma string concatenada

FUNÇÃO process_input(tamanho, número):

total_width = (comprimento do número) * (tamanho + 3) - 1

screen = matriz 2D de (2 * tamanho + 3) linhas e largura total preenchida com " "

PARA cada dígito EM número:

Obtenha os segmentos correspondentes ao dígito

Calcule start_col como (índice do dígito * (tamanho + 3))

PARA cada segmento EM segmentos:

SE segmento É top:

Preencha a linha 0 de start_col + 1 até start_col + tamanho com "-"

SENÃO SE segmento É top-left:

Preencha as linhas 1 até tamanho na coluna start_col com "|"

SENÃO SE segmento É top-right:

Preencha as linhas 1 até tamanho na coluna start_col + tamanho + 1 com "|"

SENÃO SE segmento É middle:

Preencha a linha tamanho + 1 de start_col + 1 até start_col + tamanho com "-"

SENÃO SE segmento É bottom-left:

Preencha as linhas tamanho + 2 até 2 * tamanho + 1 na coluna start_col com "|"

SENÃO SE segmento É bottom-right:

Preencha as linhas tamanho + 2 até 2 * tamanho + 1 na coluna start_col + tamanho + 1 com "|"

SENÃO SE segmento É bottom:

Preencha a linha 2 * tamanho + 2 de start_col + 1 até start_col + tamanho com "-"

CHAME show_number(screen)

INICIE PROGRAMA PRINCIPAL:

REPITA:

RECEBA entrada tamanho (s) e número (n)

SE s = 0 E n = "0":

SAIA

CHAME process_input(tamanho = s, número = n)