LISTA DE EXERCÍCIOS
ASSEMBLY RISC-V

1. Determine o comprimento de uma string no estilo C adicionando 1 a um acumulador até encontrar o terminador '\0'. A string deve ser inicializada estaticamente na área de dados globais. O programa deve mostrar o resultado com a chamada de sistema `write`.
2. Repita o exercício 1, implementando o que é pedido na forma de uma função. Teste a função passando como parâmetro a string global.
3. Escreva uma função que faça uma copia de uma string estilo C para outra. A cópia deve ser o reverso da string original. A função deve retornar o número de bytes copiados. Teste a função imprimindo a cópia usando a chamada de sistema `write`.
4. Escreva uma função que converta um inteiro (32 bits, complemento de 2) para string. Para testar a função, imprima o resultado convertido.
5. Escreva uma função para converter uma string para inteiro. O número deve ser positivo ou negativo. Para testar capture a string do teclado, usando a chamada de sistema `read` (número 63) e faça o tratamento apropriado.
6. Escreva uma função para determinar se um número inteiro é impar ou par. A função deve retornar 1 se o número par e 0 se o número for ímpar.
7. Escreva uma função que determina o número de bits 1 recebido como seu primeiro argumento (`a0`). Por exemplo, se o campo de 32 bits contiver 6 bits 1 bits, então a função deve retornar 6. Escreva um programa de teste para a função.
8. Escreva uma função que retorne a soma dos elementos de um array de inteiros. A função deve receber como parâmetros o array e o número de elementos que o compõe. A função deve imprimir o resultado na tela. Para isso use a função de conversão definida no exercício 4.
9. Escreva uma função que ordene (em ordem crescente) os elementos de um array de inteiros. A função deve receber como parâmetro o array e o número de elementos que o compõe. Use o algoritmo bubble sort para fazer a ordenação. Mostre o resultado na tela do computador usando a chamada de sistema `write`. Para isso, use a função de conversão especificada no exercício 4.
10. Implemente uma função iterativa que receba um interio n e calcule o n-ésimo número da série de Fibonacci.
11. Implemente o que é pedido no exercício 10 de forma recursiva.
12. Os exercícios da seção 6 do livro texto indicado são mais desafiadores e indicados: https://riscv-programming.org/ale-exercise-book/book/ch06-00-assembly-user-level-programming.html. Não precisa fazer exercícios que demandam alocação dinâmica.