

Lista de Exercícios – Capítulo 2

Observações:

1. Coloquem as questões em um arquivo .zip e renomeie este arquivo .zip com o seu login.
2. Façam códigos legíveis (indentados e comentados).
3. Usem o assembler do RISC-V e simulador CompSim
4. A lista é individual e cópias serão penalizadas

QUESTÕES

1. [0.5] Encontre a menor sequência de instruções do RISC-V que implementa a código abaixo, onde a, b e m são variáveis inteiras na memória. Teste com os seguintes valores de a, b e m:

| a | b |
|----|----|
| 5 | 9 |
| 10 | 10 |
| 15 | 7 |

```
int a = ;
int b = ;
int m = ;
m = a;
if ( b == m )
    m = b - a;
else
    m = a - b;
```

2. [0.5] Encontre a menor sequência de instruções RISC-V que implementa a código abaixo, onde a, b, c e x são variáveis na memória:

```
int a = ...; #qualquer valor
int b = ...;
int c = ...;
int x = ...;
x = 0;
if ( a >= 0 && b <= 32 && c > 25 )
    x = 1;
```

3. [1.0] Encontre a menor sequência de instruções RISC-V que extrai os bits 11 até 4 do registrador x10 e usa o valor desse campo para substituir os bits 27 até 20 no registrador x11 sem alterar os outros bits dos registradores x10 ou x11.

(Certifique-se de testar seu código usando x10 = 0xffff0000 e x11 = 0x0000ffff)

4. [1.0] Forneça um conjunto mínimo de instruções RISC-V que possam ser usadas para implementar a seguinte pseudo-instrução: NOT x10, x11 // inversão bit a bit

5. [1.0] Escreva um código em linguagem assembly do RISC-V que receba uma string no teclado do CompSim e retorne a quantidade de vogais da string no monitor do CompSim. Usar apenas letras minúsculas.

6. [1.5] Um estudante de graduação resolveu criar um jogo que desafiasse sua inteligência, e, para isso, ele resolveu convocar você para ajudá-lo! O jogo, que é em dupla, começa quando os jogadores 1 e 2 enviam uma palavra (string) cada pelo teclado. Após enviar, esperam para que sua pontuação seja calculada e mostrada no visor, que deverá informar o jogador que ganhou ou se houve empate.

A pontuação de cada jogador é calculada baseada na soma dos valores referentes as letras da tabela abaixo:

| Letras | Valor |
|---------------|-------|
| A, E, I, O, U | 1 |
| D, G, T | 2 |
| B, C, M, N, P | 3 |
| F, H, V, W, Y | 4 |
| K, R, S | 5 |
| J, L, X | 8 |
| Q, Z | 10 |

Exemplo:

A pontuação da palavra "GOTICO" é 10, pois:

- 2 pontos para G
- 1 ponto para O, duas vezes
- 2 pontos para T
- 1 ponto para I
- 3 pontos para C

Obs: O jogo só aceitará **letras maiúsculas**.

Obs²: Acentuação será desconsiderada (não serão enviadas palavras com acentuação ou cedilha).

7. [1.5] Um outro aluno resolveu testar suas habilidades matemáticas fazendo fatoriais. Para dificultar o treino ele resolveu somar os valores do fatorial de cada dígito de um número. Escreva um código em Assembly do RISC-V que simule o raciocínio desse aluno recebendo um número de até 6 dígitos e devolvendo a soma dos fatoriais.

Ex 1.: Entrada => 1234

Saída => 33 ($1! + 2! + 3! + 4! = 33$)

Ex 2.: Entrada => 678

Saída => 46080 ($6! + 7! + 8! = 46080$)

8. [1.5] Escreva um código em assembly do RISC-V que faça uma leitura analógica do potenciômetro do sensor de temperatura do Arduino virtual e acenda um led verde, amarelo ou vermelho dependendo do valor da leitura conforme a tabela abaixo.

| Temperatura | Led |
|--------------|--------------------|
| 0-10 | Vermelho -esquerda |
| 10- 15 | Amarelo – esquerda |
| 15 - 20 | Amarelo - direita |
| 20 - 25 | Verde - esquerda |
| 25 – 30 | Verde - direita |
| Maior que 30 | Vermelho - direita |

9. [1.5] Escreva um código que implementa um decodificador para o display de 7-segmentos do Arduino virtual. O número que aparecerá no display de 7 segmentos será de acordo com a entrada digital que tiver o valor colocado em UM, conforme a tabela abaixo.

| Entrada digital com valor em UM | Número do display 7 segmentos |
|---------------------------------|-------------------------------|
| 1 | 0 |
| 2 | 1 |
| 3 | 2 |
| 4 | 3 |
| 5 | 4 |
| 6 | 5 |
| 7 | 6 |
| 8 | 7 |
| 1 e 2 | 8 |
| 1 e 3 | 9 |
| 1 e 4 | Limpa display |