

Lista de Exercícios - Aula prática RISC V – 2019.2

Observações:

1. A lista pode ser feita em equipe (máximo 4 pessoas)
2. Os exercícios 1, 2 e 3 deverão ser entregues até às 23:59 de hoje 23/08 no link disponível no site da disciplina.
3. Os demais exercícios (4 a 6) deverão ser entregues até às 23:59 do dia 06/09/2019.
4. Coloquem as questões em um arquivo .zip ou .rar e renomeie este arquivo .zip ou .rar com o login dos integrantes dos grupos separados por underline(_). Caso queiram, coloque também um .txt com o nome completo e login dos integrantes do grupo no arquivo .zip ou .rar.
5. A entrega da lista deve ser feita **EXCLUSIVAMENTE** no link disponível no site da disciplina.
6. Façam códigos legíveis (indentados e comentados).

QUESTÕES

1. Codifique um programa correspondente ao seguinte pseudo-código:

```
int a = 5;
int b = 4;
int m = 12;
m = a;
if ( b == m )
    m = b - a;
else
    m = a - b;
```

2. Codifique um programa correspondente a:

```
int a = ...; #qualquer valor
int b = ...;
int c = ...;
int x = ...;
x = 0;
if ( a >= 0 && b <= 64 && c > 24 )
    x = 1;
```

3. (mcbs) Escreva um código em RISC-V Assembly que receba uma string e salve no registrador x18 a quantidade de vogais da string. Considere apenas letras maiúsculas.
4. (rma6) Faça um programa para RISC-V que dado uma string S (guardada na memória) guarde o valor 1 no registrador a0 caso S seja um palíndromo ou 0 caso contrário. Não diferencie maiúsculas de minúsculas e caso a string contenha espaços esses podem ser ignorados na verificação, ou seja, “Luz Azul” deve ser considerado palíndromo.

5. (lcbm2) Miguel, cansado de jogar League of Legends, resolveu criar um novo jogo que desafiasse ainda mais sua inteligência, e, para isso, ele resolveu convocar você e seu grupo para ajudá-lo!

O jogo, que é em dupla, começa quando os jogadores 1 e 2 enviam uma palavra (string) cada. Após enviar, esperam para que sua pontuação seja calculada e assim descobrir quem ganhou. Caso o jogador 1 ganhe, a saída será "**A**", por outro lado, caso o jogador 2 ganhe, a saída será "**B**". No caso de empate, a saída será "**E**".

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, N, R, S	1
D, G, T	2
B, C, M, P	3
F, H, V, W, Y	4
K	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

Sugestão: Salvar o resultado no registrador x7.

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

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

6. (rma6) A associação mundial dos monitores de hardware adora jogos para testar suas habilidades mentais e por isso inventaram um jogo cujo objetivo é descobrir quantas vezes uma dada palavra pode ser formada com um conjunto de letras. Por exemplo, dado o conjunto "xmrclszumskvbqchuwtygyieip" a palavra riscv pode ser formada 2 vezes. Como os monitores estão sempre muito atarefados eles pediram para os alunos fazerem um programa para RISC-V que resolva esse problema. O conjunto de letras e a palavra estão armazenadas na memória e o resultado deve ser guardado no registrador a0.

EXTRA. (mlpcs) **Factorion** é um número natural cujo a soma do fatorial dos seus dígitos é igual ao seu valor, por exemplo:

145 é um **factorion** em base 10 pois $1! + 4! + 5! = 1 + 24 + 120 = 145$

10 é um **factorion** em base 2 pois $1! + 0! = 10$

Escreva um programa em assembly do RISC-V que leia dois valores da memória:

o número a ser analisado e a base numérica, e indique se o número é um factorion ou não.

Obs: Se o número for factorion, salvar o valor 1 no registrador s2[x18] e salvar 0 caso contrário.