

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 e simulador do ARMv8

QUESTÕES

1. [0.5] Encontre a menor sequência de instruções LEGv8 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	5
10	3
3	10

```
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 LEGv8 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 <= 64 && c > 24 )
    x = 1;
```

3. [1.0] Encontre a menor sequência de instruções LEGv8 que extrai os bits 16 até 11 do registrador X10 e usa o valor desse campo para substituir os bits 31 até 26 no registrador X11 sem alterar os outros bits dos registradores X10 ou X11.

(Certifique-se de testar seu código usando X10 = 0 e X11 = 0xffffffffffff)

4. [1.0] Forneça um conjunto mínimo de instruções LEGv8 que possam ser usadas para implementar a seguinte pseudo-instrução: NOT X10, X11 // inversão bit a bit
5. [1.0] Considere uma nova instrução proposta chamada RPT. Esta instrução combina a verificação de condição de um loop e o decréscimo do contador em uma única instrução. Por exemplo, RPT X12, loop faria o seguinte:

if (X12 > 0) {X12 = X12-1; go to loop}

- a. Se esta instrução fosse adicionada ao conjunto de instruções do ARMv8, qual é o formato de instrução mais apropriado?
 - b. Qual é a sequência mais curta de instruções LEGv8 que executa a mesma operação?
6. [1.0] Escreva um código em linguagem assembly do LEGv8 que receba uma string e retorne a quantidade de vogais da string. Usar apenas letras maiúsculas.
7. [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. 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

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

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

8. [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 LEGv8 que simule o raciocínio desse aluno recebendo um número de até 5 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)

9. [2.0] Um grupo de alunos do CIn adora jogos para testar suas habilidades mentais e por isso inventaram um jogo cujo objetivo é descobrir quantas vezes uma palavra pode ser formada com um conjunto de letras. Por exemplo, dado o conjunto “gnrclszemskvbgcluwtygyvieip” a palavra leg, o programa retorna que a palavra pode ser formada 2 vezes. Faça um programa em assembler do LEGv8 que resolva esse problema. O conjunto de letras contém no máximo 20 letras e a palavra são entradas do programa.