

Lista 1
Aula prática Assembly MIPS 2022.2
Ciência da Computação
Aula prática assembly MIPS

Atenção: o código de todas as questões deverá estar claramente comentado. Em caso contrário, a correção será muito dificultada e é de seu interesse colaborar com a correção!

Obs: os exercícios 1, 2 e 3 deverão ser entregues no mesmo dia da aula na atividade do Classroom (apenas um arquivo zip ou tar.gz por grupo com os arquivos .asm/txt de cada questão e um arquivo .txt com o número da lista e integrantes do grupo).

Os demais exercícios (4 a 11) deverão ser entregues até a data marcada no calendário como entrega da lista (apenas um arquivo zip ou tar.gz por grupo com os arquivos .asm/txt de cada questão e um arquivo .txt com o número da lista e integrantes do grupo).

1. (0,75) Codifique um programa correspondente ao seguinte pseudo-código:

```
int a = ?; //input
int b = ?; //input
int result; // Posição na memória

if(a < b)
    result = a;
else if(a > b)
    result = b;
else
    result = a+b;
```

2. (0,75) Codifique um programa correspondente ao seguinte pseudo-código:

```
int a = ?; // input a >= 0 && a <= 100
int b = 0;
int result_perfect_square; //posição na memória
int result_not_perfect; //posição na memória
for (int i = 0; i != 10; i++) {
    if (i * i == a) {
        b = 1;
        result_perfect_square = a; //guarda a na
        memória
```

```

        break;
    }
}
if (b == 0) result_not_perfect = a; //guarda a na memória

```

3. Responda:

- a. (0,5) Qual a diferença entre uma arquitetura RISC e uma CISC?
- b. (0.75) Explique o propósito da arquitetura MIPS

4. (0,5) Codifique um programa correspondente ao seguinte pseudo-código:

```

int b = 2;
int e = ?; //input e >= 1 && e <= 5
int res = 1;

for(int i = 0; i < e; i++) {
    res *= 2;
}

```

5. (0,6) Escreva um programa em linguagem de montagem do MIPS que receba três números inteiros armazenados na memória e verifica se aqueles três números podem ser comprimentos de lados de um triângulo, e caso sejam, defina qual tipo de triângulo ele é, armazenando o resultado numa string s na memória, “not” caso não sejam comprimentos de lados de um triângulo, “eq” para equilátero, “iso” para isósceles, “esc” para escaleno. Lembrando que para três números serem os comprimentos dos lados de um triângulo, cada lado deve ser menor que a soma dos outros dois.

6. (1,4) Escreva um programa em linguagem de montagem do MIPS que receba dois números inteiros armazenados na memória e realiza a divisão inteira dos dois números. Considere números positivos e negativos. A instrução “div” não deverá ser utilizada na implementação dessa questão. O resultado (quociente da operação) deverá ser armazenado em uma variável RESULT na memória e o resto da divisão deve ser armazenado em uma variável REMAINDER na memória.

7. (0,6) Desenvolva um programa que recebe como entrada um array de inteiros a e escreve em um outro array b todos os números que são cubos perfeitos.

Ex.: a = [1,15,27,88,125,138] → b = [1,27,125]

Obs.: Assuma que só serão testados números no intervalo [0, 10000].

8. (1,25) Implemente (recursivamente) na linguagem de montagem do MIPS,

uma função que receba dois números (a e b) e retorne a mod b. Se a for negativo, armazene o valor 1 no registrador v1 e encerre o programa.

9. (1,4) Leia uma string da memória, transforme as vogais em maiúsculas, as consoantes em minúsculas e ordene as letras em vogais primeiro e consoantes depois na mesma ordem de aparição e salve-os (já ordenados) no mesmo lugar da memória. Se houver algum caractere que não seja letra, o valor 1 deverá ser armazenado no registrador v1 e o programa encerrado. Exemplo:

entrada - "enTRadA"

saída - "EAAntrd"

10. Responda:

a. (0,3) Qual a diferença entre Arquitetura e Organização de um computador?

b. (0,7) Quais as principais diferenças entre utilizar registradores e memória?

c. (0,5) Qual a diferença entre Big Endian e Little Endian? E como são armazenados na memória?

11. (0,0) Considere o seguinte pseudocódigo:

```
char *ptr = ?; /* input ponteiro para código mips
               assembly representado como array de char */
if (halts(ptr)) {
    while (1 == 1) {
        continue;
    }
} else {
    return;
}
```

Implemente a função halts() que recebe um ponteiro para um código arbitrário em assembly mips e retorna 1 caso a execução do código pare eventualmente ou retorna 0 caso a execução entre em loop.

Boa Sorte!