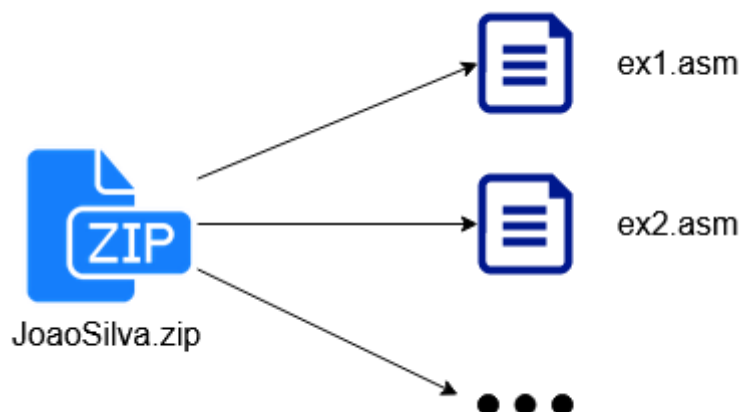


## Lista de Exercícios do MARS

Essa lista de exercícios vale nota e sua resolução deverá ser entregue no Microsoft Teams até o dia **06/04** às **23:55**. A entrega poderá ser feita **individual ou em dupla**, ficando a critério dos alunos resolver a lista sozinho ou não, sendo que no caso da dupla apenas um dos integrantes precisa submeter. Deverá ser submetido um arquivo zip contendo a resolução dos exercícios. O arquivo zip deverá ter o nome do aluno ou da dupla (ex: JoaoSilva.zip). Para cada exercício da lista deverá ser submetido um arquivo com extensão ASM (esses arquivos deverão estar dentro do zip). Os arquivos deverão ser nomeados conforme o nome do exercício (ex: ex1.asm, ex2.asm...). Dentro de cada arquivo deverá constar um cabeçalho com o nome e o RA do aluno (ou dos dois alunos da dupla) na primeira linha do código. Para cada linha de código assembly escrita também deverá ser apresentado um comentário descrevendo o que a linha está fazendo.

Dúvidas sobre o entendimento do parágrafo acima podem ser perguntadas a qualquer momento antes do encerramento da atividade. A professora e os monitores também estão disponíveis para tirar dúvidas sobre os exercícios.



1) Faça um código em assembly do MIPS que realize a soma de todos os dígitos do seu RA. Por exemplo, se seu RA fosse 12345, a soma seria  $1+2+3+4+5=15$ . OBS: Para as duplas, a soma dos dígitos do RA poderá ser realizada com o RA de qualquer um dos dois integrantes da dupla.

2) Faça um código em assembly do MIPS que dados dois números, armazenados em t0 e t1, calcula o maior destes números e o armazena em t5

3) Transforme o seguinte código descrito com linguagem C em assembly do MIPS:

```
if (x == y)
    z = 2;
else
    z = 3;
```

x deve ser armazenado em t0

y deve ser armazenado em t1

z deve ser armazenado em t5

Escolha valores arbitrários para x e y, e não se esqueça de testar ambos os casos

4) Transforme o seguinte código descrito com linguagem C em assembly do MIPS:

```
x = 8;
y = 4;
i = 2;
while(i > 0){
    y = y + 2;
    i = i - 1;
}
z = x + y;
```

x deve ser armazenado em t0

y deve ser armazenado em t1

i deve ser armazenado em t2

z deve ser armazenado em t5

5) Faça um código em assembly do MIPS que dado um número, armazenado em t0, calcula o fatorial deste número e o armazena em t5. (Dica: você pode utilizar a instrução de multiplicação do MIPS (mul)).

6) Faça um código em assembly do MIPS que dados dois números, armazenados em t0 e t1, implementa o **algoritmo de multiplicação por somas sucessivas**. O resultado da multiplicação deverá ser armazenado em t5. Para facilitar a implementação, considere que os números sempre serão números positivos.

7) Um número “a” é considerado um quadrado perfeito quando é possível obter este número a partir do quadrado de um outro número ( $n^2 = a$ ). Faça um código em assembly do MIPS que dado um número, armazenado em t0, verifica se o número é um quadrado perfeito. Caso ele seja, deverá ser armazenado 1 em t5, caso contrário, deverá ser armazenado 0 em t5 (Dica: elevar um número ao quadrado é a mesma coisa que multiplica-lo por ele mesmo).

8) Faça um código em assembly do MIPS que dados dois números, armazenados em t0 e t1, calcula o resto da divisão de t0 por t1 ( $t0/t1$ ). Este resto da divisão deverá ser armazenado em t5 (OBS: É possível obter esse valor a partir de algumas rotinas/funções do MIPS, mas isso **não** deverá ser feito aqui). Para facilitar a implementação, considere que os números sempre serão números positivos.