

# AOC-1: Trabalho Prático 5

## ► Instruções

- Use apenas instruções vistas em aula até agora (slides)
- No MARS, use a seguinte configuração:
  - No menu *Settings*, desabilite a opção *Permit extended (pseudo) instructions and formats* e
  - **HABILITE A OPÇÃO *DELAYED BRANCHING***
- Seus exercícios serão corrigidos com o MARS configurado da forma descrita acima
- Utilize **EXATAMENTE** os registradores explicitados nos exercícios
  - Resultados armazenados em registradores diferentes serão considerados incorretos
- Comente seu código
- Todos os exercícios são individuais
  - Cópias detectadas resultarão em nota zero para ambos os alunos

# AOC-1: Trabalho Prático 5

## ► Instruções

- O material deve ser entregue pelo AVA (<http://ava.ufpel.edu.br>) e deverá obedecer às seguintes regras:
  - Será um arquivo compactado (**obrigatoriamente** no formato **.zip**) contendo os códigos fonte dos TPs
    - Ex: *fulano\_da\_silva.zip*
  - Cada exercício deve ter o seguinte nome:
    - *matricula\_tp{n}\_e{m}.asm*
    - Onde:
      - *matricula* é a matrícula do aluno e
      - *{n}* é o número do TP
      - *{m}* é o número do exercício
      - Ex: *16100001\_tp1\_e1.asm*, *16100001\_tp1\_e2.asm*, ...
- Trabalhos que não seguirem as regras acima **NÃO SERÃO CORRIGIDOS!**
- O prazo de submissão do trabalho é até as **23:55** da próxima **terça-feira**.  
**NÃO** serão aceitos exercícios após a data/hora-limite

# AOC-1: Trabalho Prático 5

## ► Instruções

1. Escreva um programa que leia dois *halfwords* (*a* e *b*) da memória e calcule a sua divisão se os dois valores forem diferentes e a sua multiplicação se os dois valores forem iguais. Escreva o resultado (*y*) em uma palavra (4 bytes) de memória. O resultado deve ser armazenado, obrigatoriamente, na posição **0x10010004** da memória `.data`. Inicie o código com:

```
.data
    a: .half 30
    b: .half 5
    y: .space 4
```

# AOC-1: Trabalho Prático 5

## ► Instruções

2. Escreva um programa que calcule:

$$1 + 2 + 3 + \dots + 333$$

Escreva o resultado (**y**) em uma palavra (4 bytes) de memória. O resultado deve ser armazenado, obrigatoriamente, na posição **0x10010000** da memória .data. Inicie o código com:

```
.data  
y: .space 4
```

# AOC-1: Trabalho Prático 5

## ► Instruções

3. Escreva um programa que leia um valor  $x > 0$  da memória (posição 0x10010000) e calcule o  $x$ -ésimo termo da série de Fibonacci:

1, 1, 2, 3, 5, 8, 13, 21, 34, 55, ...

Escreva o  $x$ -ésimo termo da série ( $y$ ) em uma palavra (4 bytes) de memória. O resultado deve ser armazenado, obrigatoriamente, na posição **0x10010004** da memória `.data`. Inicie o código com:

```
.data
x: .word 7
y: .space 4
```

# AOC-1: Trabalho Prático 5

## ► Instruções

4. Uma temperatura, armazenada em **\$t0**, pode ficar em dois intervalos:

$20 \leq \text{temp} \leq 40$  e

$60 \leq \text{temp} \leq 80$ .

Escreva um programa que coloque uma flag (registrador **\$t1**) para 1 se a temperatura está entre os valores permitidos e para 0 caso contrário.

Inicie o código com a instrução: **ori \$t0, \$zero, temperatura**, substituindo *temperatura* por um valor qualquer.