

Lei atentamente as instruções abaixo:

1. Esta atividade deve ser realizada individualmente.
2. A atividade consiste na implementação de **três exercícios** utilizando procedimentos na linguagem de montagem do MIPS (conforme as instruções a seguir).
3. Deve ser postado um relatório, com uma capa identificando a Instituição, o curso, a disciplina, o professor, o nome da atividade, os autores do trabalho e a data em que o mesmo for entregue.
4. O corpo do relatório deverá conter a resolução dos exercícios, incluindo: **código-fonte em linguagem de alto nível** (preferencialmente C ou C++), **código-fonte em linguagem de montagem do MIPS** e **capturas de tela** que ***demonstrem claramente a execução correta das entradas e saídas realizadas via console do simulador e os resultados da execução dos programas***. Para cada exercício desenvolvido, apresentar uma discussão dos resultados.
5. Cada código fonte deve conter um cabeçalho comentado que identifique a disciplina, a atividade, o programa e o nome do aluno. Ex:


```
# Disciplina: Arquitetura e Organização de Processadores  
# Atividade: Avaliação 03 – Procedimentos em Linguagem de Montagem  
# Exercício 01  
# Aluno: Ringo Starr
```
6. O relatório deve ser em formato PDF e postado no ambiente Material Didático, conforme instruções fornecidas em aula. O prazo para entrega do relatório é o indicado no ambiente Material Didático. Não serão aceitos trabalhos entregues em atraso.
7. **A implementação deverá apresentar resultados corretos para qualquer conjunto de dados.** Uma solução que **não execute corretamente** terá, automaticamente, um **desconto de 50% na nota**, sendo que o professor também avaliará a correção de segmentos específicos do código (controle de execução, acesso a memória,...).
8. Se forem identificados **trabalhos** com grau de **similaridade** que caracterize cópia (autorizada ou não) ou adaptação, a nota dos grupos será a **nota de um trabalho dividida** pelo número de grupos que entregou esses trabalhos similares.

Exercício 01 – Chamada de procedimento

Implemente um programa que:

- Leia dois números (X e Y) usando a instrução `syscall`;
- Chame um procedimento chamado SOMA, passando X e Y nos registradores de argumento;
- Imprima o valor retornado pelo procedimento, o qual deve fazer a soma de X e Y.

O código do procedimento deve ser escrito no início do segmento de código (`.text`), antes da função principal (`main`). No entanto, no MARS, o procedimento deve ser precedido por um desvio incondicional para a função principal. Exemplo:

```
.text
j main
soma : ...
      jr $ra

main : ....
      jal soma
```

OBS: Este exercício aborda o uso das instruções `jal` e `jr`.

Exercício 02 – Uso de pilha

Implemente um programa que:

- Inicialize os registradores `$s0`, `$s1` e `$s2` com os respectivos valores 7, 8 e 9;
- Estenda o procedimento anterior (Exercício 01) de tal forma a:
 - Salvar `$s0`, `$s1` e `$s2` na pilha;
 - Executar as seguintes operações:

```
$s0 <- $a0
$s1 <- $a1
$s2 <- $a2
```

- Recupere os valores salvos na pilha e retorne com a soma.

- Imprima o valor retornado pelo procedimento

Exercício 03 – Vetor

Implemente um programa que:

- Solicite a entrada de um vetor com oito (8) elementos;
- Chame um procedimento fornecendo o endereço base e o tamanho do vetor;
- Imprima o valor retornado pelo procedimento que deve conter o número de elementos nulos (zero) no vetor. Exemplo:

```
Vetor_A: .word 9 0 7 0 3 0 4 1
```

- Para o vetor apresentado o resultado do programa deverá ser 3, ou seja, o vetor A possui 3 elementos nulos (iguais a zero).