

### **Instruções:**

1. Esta atividade pode ser realizada individualmente ou em grupo de **até DOIS** alunos (no máximo).
2. A atividade consiste na implementação de um procedimento utilizando a linguagem de montagem do MIPS (conforme as instruções a seguir), o qual é apresentado logo a seguir.
3. Deve ser entregue um relatório impresso em papel, com uma capa identificando a Instituição, o curso, a disciplina, o professor, o nome da atividade, os autores do trabalho e data em que o mesmo for entregue. O corpo do relatório deverá conter a resolução dos exercícios, incluindo: **código-fonte** em linguagem de montagem e **capturas de tela** demonstrando a execução correta das entradas e saídas realizadas via **console do simulador**.
4. O código-fonte deve ser escrito em arquivo com extensão .asm e, **obrigatoriamente**, com um nome que identifique os membros do grupo (ex. ***RingoStarr\_JohnLennon.asm***). Obs: Todos os arquivos de todos os grupos serão reunidos em uma mesma pasta e, por isso, não use nomes como Programa\_01.asm.
5. Cada código fonte deve conter um cabeçalho comentado que identifique a disciplina, a atividade, o programa e os nomes dos membros do grupo. Ex:  

```
# Disciplina: Arquitetura e Organização de Computadores  
# Atividade: Avaliação 03 - Programação de Procedimentos  
# Grupo: - Ringo Starr  
#         - John Lennon
```
6. O arquivo ASM e o relatório em formato PDF deverão ser postados no ambiente Material Didático compactados em um único arquivo em formato ZIP, conforme instruções fornecidas em aula.
7. O prazo para entrega do relatório e postagem dos códigos fonte é 19h do dia **06/10/17**.
8. Não serão aceitos trabalhos entregues em atraso.
9. O professor poderá solicitar a qualquer momento que **qualquer aluno** do grupo faça uma **demonstração explicativa da execução dos códigos no MARS**.
10. **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,...).
11. 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.

## ESPECIFICAÇÃO DO PROBLEMA

---

### Enunciado:

Utilizando a linguagem de montagem do MIPS, implemente um procedimento que determine o máximo divisor comum entre dois números passados como parâmetros ao procedimento.

### Requisitos:

1. A função principal (MAIN) deve realizar a leitura dos dois números através da entrada do console, copiá-los para registradores de argumento e chamar o procedimento.
2. No retorno do procedimento, a função principal deve imprimir uma mensagem informando o valor do máximo divisor comum calculado pelo procedimento.
3. O procedimento deve utilizar pelo menos um registrador \$s.
4. A função principal e o procedimento devem respeitar as regras relativas ao uso dos registradores do MIPS, preservando na pilha os registradores de acordo com o especificado nessas regras (ex. a função principal deve preservar os registradores \$t utilizados e o procedimento deve preservar os registradores \$s). Verifique as regras descritas no livro texto.
5. O código deve ser escrito respeitando o estilo de programação ASM, usando tabulação para organizar o código em colunas (rótulos, mnemônicos, operandos e comentários).
6. Procure comentar ao máximo o seu código. Isto é um hábito da programação *assembly*.
7. No Apêndice A do livro estão listadas todas as instruções e pseudo-instruções do MIPS, incluindo diferentes instruções úteis para comparação de valores e desvio condicional.
8. No seu relatório, apresente uma análise indicando quais instruções do conjunto de instruções do MIPS foram utilizadas.

---

### NOTA:

**Abaixo, segue uma referência de código C de um procedimento que implementa o algoritmo do máximo divisor comum:**

```
int proc_mdc(int x, int y) {
    while (x != y) {
        if (x < y)
            y = y - x;
        else
            x = x - y;
    }
    return x;
}
```

---