

AOC-1: Trabalho Prático 1

► Instruções

- Use apenas instruções vistas em aula até agora (slides)
- No MARS, use a seguinte configuração:
 - No menu *Settings*, desabilite as opções
Permit extended (pseudo) instructions and formats e
Delayed Branching
- Seus exercícios serão corrigidos com o MARS configurado da forma descrita acima
- Comente seu código
- Todos os exercícios são individuais
 - Cópias detectadas resultarão em nota zero para ambos os alunos
- Submeta os exercícios no AVA em um arquivo compactado até as 23:55 de **quinta-feira**, 20/08/2015. Não serão aceitos exercícios após a data/hora-limite.

AOC-1: Trabalho Prático 1

► Instruções

1. Faça um programa que escreva o valor 0xDECADEA70 no registrador \$t7, incluindo um dígito hexadecimal por vez (isto é, insira letra por letra, individualmente) no registrador.
2. Escreva 0xAAAAAAAA em \$t1. Faça um *shift* de um bit para a direita desse valor e coloque o resultado em \$t2, deixando \$t1 inalterado. Em \$t3, \$t4 e \$t5 coloque os resultados das operações or, and e xor entre \$t1 e \$t2, respectivamente. Em comentários no final do código, explique os resultados obtidos, mostrando os valores binários.

AOC-1: Trabalho Prático 1

► Instruções

3. Comece um programa pela instrução: `ori $t1, $zero, 0x01`. Usando apenas as instruções lógicas do tipo R (dois registradores como operandos) `or`, `and` e `xor` e instruções de *shift* (isto é, sem usar outras instruções `ori` além da inicial e nem instruções `nor`), escreva `0xFFFFFFFF` em `$t1`. Procure usar o menor número de instruções possível.
4. Escreva a sequência `0x12345678` em `$t1`. A seguir, escreva um código que inverta a sequência, escrevendo `0x87654321` em `$t2`. Obviamente o código deve inverter os bits de `$t1` e não simplesmente escrever `0x87654321` diretamente em `$t2`.