

AOC-1: Trabalho Prático 8

► Instruções

- Use apenas instruções vistas em aula até agora (slides)
- No MARS, use a seguinte configuração:
 - No menu *Settings*, **habilite** 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 8

► 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 8

► Instruções

1. Escreva um programa que repetidamente pergunte ao usuário pelo número de quilômetros viajados e litros de gasolina consumidos e depois imprima o número de quilômetros por litro. Para sair do programa, o usuário deve digitar 0 como número de quilômetros.

Armazene todos os números de quilômetros por litro na memória, iniciando pelo endereço 0x10010000.

Exemplo:

Entrada	Saída	
40		
3	13	-- 13 também deve ser armazenado em 0x10010000
60		
5	12	-- 12 também deve ser armazenado em 0x10010004
0		-- Sai do programa

AOC-1: Trabalho Prático 8

► Instruções

2. Escreva um programa que leia uma string (código 8) representando um inteiro positivo e converta a string para um inteiro. Armazene o inteiro no registrador **\$t0**. Assuma que a entrada conterá apenas dígitos (0-9). Verifique sua resposta imprimindo o inteiro correspondente (código 1). Exemplo:

Entrada
"3452"

Saída
3452

-- 3452 também deve ser armazenado em **\$t0**

AOC-1: Trabalho Prático 8

► Instruções

3. Escreva uma subrotina que retorne a média entre três valores. A subrotina deve receber como argumentos três inteiros e retornar a média entre eles. Escreva um programa para testar a subrotina que leia três valores do usuário e imprima o resultado. Importante: Utilize os registradores convencionados para a passagem de argumentos e valor de retorno! Armazene o retorno da subrotina (média entre os três valores) no registrador **\$t5**.

Exemplo:

Entrada	Saída	
51		
42		
39	44	-- 44 também deve ser armazenado em \$t5

AOC-1: Trabalho Prático 8

► Instruções

4. Escreva um programa que leia um vetor da memória e execute o algoritmo de ordenação por inserção (insertion sort). O programa deve imprimir o vetor a cada passo da execução do algoritmo (após cada inserção). Codifique o algoritmo em um procedimento que receba dois parâmetros: o endereço do primeiro elemento do vetor (**\$a0**) e seu tamanho (**\$a1**). Armazene o vetor ordenado na memória, após o vetor de entrada. Por exemplo, para o vetor **3, 7, 4, 9, 5, 2, 6**, a saída deve ter o seguinte formato:

```
3 7 4 9 5 2 6
3 7 4 9 5 2 6
3 4 7 9 5 2 6
3 4 7 9 5 2 6
3 4 5 7 9 2 6
2 3 4 5 7 9 6
2 3 4 5 6 7 9
```