UNIVALI – Universidade do Vale do Itajaí Disciplina – Arquitetura e Organização de Processadores
Avaliação 04 – Programação de procedimentos em linguagem de montagem

Lei atentamente as instruções abaixo:

1. Esta atividade pode ser realizada em até 3 pessoas.

2. A atividade consiste na implementação de um programa 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 do problema, 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 os nomes dos membros do grupo. Ex:

```
# Disciplina: Arquitetura e Organização de Processadores
# Atividade: Avaliação 04 – Programação de Procedimentos
# Alunos: Ringo Starr, John Lennon, Paul McCartney
```

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.

Enunciado:
Utilizando a linguagem de montagem do MIPS, implemente um procedimento para ordenação de vetores baseado no algoritmo da bolha (bubblesort) – ver em http://pt.wikipedia.org/wiki/Bubble_sort.

Requisitos:

1. **Declaração do vetor:** Na seção de declaração de variáveis (.data), o vetor deve ser declarado com 8 elementos inicializados em 0, e ser claramente identificado como Vetor_A, por exemplo.

2. **Entrada do tamanho dos vetores:** A função principal (MAIN) deve solicitar o número de elementos do vetor aceitando no máximo um vetor com 8 elementos. Para leitura, deve ser apresentada uma mensagem solicitando a entrada desse valor, indicando o seu limite máximo.

3. **Verificação do tamanho do vetor:** A função principal (MAIN) deve solicitar a entrada do número de elementos até que ele seja maior que 1 e menor ou igual a 8. No caso de entrada inválida, a função principal deve imprimir uma mensagem de advertência antes de solicitar novamente a entrada.

4. **Entrada dos elementos dos vetores:** Para leitura, a função principal (MAIN) deve solicitar ao usuário a entrada de cada elemento do vetor, um a um, com mensagens do tipo:
   Vetor_A[0] =
   Vetor_A[1] =
   ...
5. **Interface da função principal com o procedimento:** A função principal (MAIN) deve copiar o tamanho do vetor para o registrador $a0 e o endereço base do vetor para o registrador $a1 antes de chamar o procedimento BUBBLESORT.

6. **Funcionalidade do procedimento:** O procedimento BUBBLESORT deve realizar a ordenação e retornar o vetor ordenado à função principal.

7. **Uso de registradores e da pilha pelo procedimento:** Caso seja necessária a utilização de algum registrador de uso geral (ex. para implementar laços de repetição), o procedimento deve, OBRIGATORIAMENTE, utilizar apenas registradores salvos $s, preservando-os na pilha no começo do procedimento e restaurando-os ao seu final.

8. **Retorno do procedimento:** Como o procedimento recebe o endereço base do vetor, não há necessidade de retornar nenhum valor pelos registradores **$v**. A função principal terá acesso ao vetor ordenado pelo registrador **$a1**.

9. **Impressão do vetor ordenado:** Após o retorno do procedimento, a função principal MAIN deve imprimir o vetor ordenado no terminal.

10. **Estilo de codificação:** O código deve ser escrito respeitando o estilo de programação ASM, usando tabulação para organizar o código em colunas (rótulos, mnemônicos, operandos e comentários). Procure comentar ao máximo o seu código.

NOTA: Para ter certeza do funcionamento correto do procedimento em qualquer circunstância, durante o seu teste, é recomendável que a função principal inicie todos os registradores $s utilizados pelo procedimento com um valor diferente de 0, como, por exemplo:

```
addi $s0, $zero, 0xFFFFFFF0
addi $s1, $zero, 0xFFFFFFF1
addi $s2, $zero, 0xFFFFFFF2
...
```

Após a execução do procedimento, se ao retornar à função principal, os valores originais desses registradores tiverem sido restaurados, então o tratamento da pilha estará correto.