Paradigmas de Linguagens de Programação (02.509-7)
Semestre 1 - 2006
Lucia Helena Machado Rino

Tema: LISP

Atenção: fazer em grupo de DOIS alunos

Considerando as funções do LISP vistas em sala de aula para iteração (DO, LOOP, etc.) e a definição de funções recursivas com mesmo fim, que, por sua vez, devem fazer uso de instruções condicionais (CASE, COND, IF, etc.) para promover a interrupção do ciclo,

PEDE-SE:

1. Eleja um problema de seu interesse, para explorar, **que envolva repetições de blocos de instruções**. Pode ser um problema já resolvido em disciplinas anteriores (inclusive com código imperativo).
2. Construa duas funções em LISP para resolver o problema eleito: UMA ITERATIVA E UMA RECURSIVA. Essas funções *devem ser equivalentes*.
3. Implemente suas funções, mostrando *resultados automáticos variados*, de acordo com o domínio suposto de resolução.
4. Compare as implementações de seu código, discutindo legibilidade, clareza e dificuldade de construir o código-fonte. Neste caso, discuta a dificuldade considerando a necessidade de verificar a consistência de seu código. Por exemplo, a resolução iterativa é mais simples ou mais complicada para esse objetivo?

OBS.: Você terá que explorar as funções do seu dialeto, para definir as suas. Elas não necessariamente foram detalhadas em aula.

DATA DE ENTREGA: 07/06/2006 (3a. feira), ATÉ 17:30 horas

LOCAL DE ENTREGA:

- **MOODLE:** *SOMENTE O CÓDIGO-FONTE*
- **RELATÓRIO:** *NO MEU ESCANINHO*

ATENÇÃO: REGISTRE ESSA FORMA DE ENTREGA!
NÃO QUEIRA FICAR CONFIRMANDO REPETIDAMENTE.
As instruções estão CLARAS, correto?

Material a ser entregue: IDENTIFICADO **ADEQUADAMENTE**: GRUPO, componentes do grupo, **TURMA** e IDENTIFICAÇÃO DO TRABALHO (T2, neste caso).

1. Documentação EM PAPEL, discutindo seu código da seguinte forma:
   1.1. Descrição BREVE do problema em foco, com inclusão do código construído – *exatamente o que foi EXECUTADO*. Este código deve ter a devida documentação interna.
   1.2. metodologia de resolução do problema
   1.3. Decisões de projeto (estruturas de dados, operações sobre dados, etc., quando aplicável ou obscuro no código)
2. Resultados de implementação
   2.1. Exemplos de execução **devem ser DUMPs DE TELA**, ou seja, devem ser resultados REAIS, gerados automaticamente. **Não reproduza** em editor de texto o resultado automático.
   2.2. mostrar exemplos de execução para casos VARIADOS, se o problema configurar alternativas de solução.
   2.3. NÃO repita resoluções similares nos exemplos.
   2.4. quando pertinente, discutir as próprias variedades de resolução.
3. Discussão das características de sua resolução. Neste caso, supõe-se que você deva
   3.1. comparar a forma de representação escolhida por você com outras formas possíveis
   3.2. ser capaz de apontar os mecanismos de controle específicos

4. Indique seu grau de dificuldade na execução deste trabalho.

Seja SUCINTO, NÃO apresente QUANTIDADE, MAS QUALIDADE

IMPORTANTE: Trabalhos entregues com uma semana de atraso terão 25% de desconto na nota. Após uma semana, não valerão mais. Não insista.

IMPORTANTE: Trabalhos sem identificação ficarão pendentes, sem avaliação.

Critérios de avaliação - não exclusivos, sujeitos a ajustes (notas máximas em cada caso):

1. Ausência de código-fonte e de resultados processados automaticamente – nota ZERO
2. Resultado não analisado e/ou duvidoso – nota máxima = 3.0
3. Resultado ruim/obscuro – nota máxima = 5.0
4. Documentação ruim ou nula – 5.0
5. Documentação média – 7.0
6. Documentação (muito) boa e discussão ok – 10.0

NOTAS ADICIONAIS

- A documentação deverá incluir visão CRÍTICA do problema. Se houver casos de resultados inesperados, por exemplo, esses devem ser discutidos.
- Documentação não deve ser CÓPIA de livro, nem reprodução de enunciado.
- Considere SOMENTE o que é específico de sua resolução, para incluir nessa documentação.