

Organização de Computadores – DEI/ISEP

Exame da parte prática

19 de Junho de 2006 – 1h30m

Conselhos para o exame:

- Tentar não usar expressões `case` ou `if`
- Dividir a definição das funções em várias cláusulas
- Usar *pattern matching* e/ou *guards* para seleccionar uma cláusula da função
- Manter o corpo da cláusula pequeno e simples
- Usar sub-funções
- Ler as duas páginas do enunciado

Avisos:

- Nota mínima na parte prática: 40%
- Todas as perguntas valem 1/3 da nota da parte prática
- Resolver cada exercício numa folha (**folha!**) separada devidamente identificada
- O teste deve ser resolvido individualmente
- É permitida a consulta de quaisquer documentos em papel que pertençam ao próprio
- Não é permitido o uso de computadores

Perguntas:

1. Faça uma função chamada `fnome/1` que aceite como argumento uma lista (uma string) que represente um nome e formate esse nome de modo a que primeiro apareça o último nome em maiúsculas, seguido por uma vírgula e depois os primeiros nomes.

Entrada:	Saída:
"José Silva"	"SILVA, José"
"José da Silva"	"SILVA, José da"
"Vicky"	"VICKY"
"Manuel da Silva e Costa"	"COSTA, Manuel da Silva e"

Nota: As strings são representadas em Erlang como listas com o código ASCII dos caracteres, encontrando-se no slide 61 dos apontamentos sobre as extensões ao Erlang uma função que transforma um carácter (qualquer que seja) numa maiúscula se tal for possível.

2. Faça um processo registado com o nome `trinco`, simulando uma fechadura electrónica com 4 dígitos cujo código é passado como argumento e que aceita as seguintes mensagens dando as seguintes respostas:

Mensagem:	Resposta:
<code>{digito, Digito, Pid}</code>	ok
<code>{digito, Digito, Pid}</code>	aberto, ou errado (no 4º dígito)

Se houver uma tentativa errada a fechadura não aceita mensagens durante os próximos 5 minutos (deve deitar fora todas as mensagens recebidas entretanto).

Se não receber mensagens (com dígitos durante 1 minuto) volta a ficar à espera do primeiro dígito.

Nota: O código para abrir a fechadura é enviado numa sequência de 4 mensagens, contendo cada uma um dígito. O resultado de sucesso ou insucesso só é enviado depois de recebido o último dígito, mesmo que qualquer dos outros dígitos esteja errado.

3. Um servidor registado com o nome `as` – *Agência de segurança* fornece *guarda-costas* aos seus processos clientes. Supondo que a função de cada *guarda-costas* é `gc:start(Pid)` em que `Pid` é a identidade do cliente, e que esta função já está escrita, implemente o servidor de modo a que:

- Sempre que o processo `as` recebe uma mensagem `{pedido, Pid}`, cria 3 *guarda-costas* para o cliente em questão.
- Sempre que um *guarda-costas* morrer os seus dois colegas devem ser terminados, ficando o cliente com três novos *guarda-costas*.
- Se o cliente morrer os *guarda-costas* devem ser terminados para que a reputação comercial da firma não saia prejudicada.
- A detecção da morte dos processos deve ser da responsabilidade do processo `as`.