

Programação Imperativa – EI e CC (1º ano)

Teste final

22 de Junho de 2011 (14h00)

Dispõe de **2:00 horas** para realizar este teste.

Questão 1 (cálculo numérico, 4v)

Codifique em C uma função que recebe dois argumentos do tipo inteiro, a e b , e imprime o somatório dos múltiplos de 3 que não são múltiplos de 2 existentes no intervalo fechado $[a, b]$.

Apresente definições para duas versões desta função:

- `int somait(int a, int b)` — que realiza o cálculo de forma iterativa;
- `int somarec(int a, int b)` — que realiza o cálculo de forma recursiva.

Questão 2 (arrays/strings, 4v)

Codifique em C uma função `char * pcript(char f[])` que recebe uma frase e a retorna criptografada na maldita linguagem dos "p's".

Nessa linguagem, cada vogal original é repetida intercalando-se um "p" no meio das vogais repetidas.

Exemplo: a frase "Oh pa, esta e uma boa frase" ficaria "Opoh papa, epestopa epe upumapa bopoapa frapasepe".

Nota: lembre-se de alocar memória que seja garantidamente suficiente para o valor a retornar.

Questão 3 (ordenação, 4v)

Desenvolva um programa em C que irá ler N números inteiros, ordená-los por ordem decrescente e escrevê-los, um por linha, no ficheiro "output.txt".

Questão 4 (modelação de estruturas de dados, 4v)

Maria e Hipolito Zacarias sentam-se à mesa de jogo para jogar o "King" com um casal amigo. Depois de uma longa discussão (como é habitual) lá assentaram nas regras para a 1ª volta do jogo, designada por "não fazer vasas". Recordando que 1 vasa é 1 conjunto de 4 cartas, essa 1ª volta consiste no seguinte: depois de baralhar as 52 cartas (13 de cada naipe), distribuem-se 13 a cada jogador formando a "mão" de cada um e não sobrando nenhuma no baralho; em cada jogada, um jogador começa (agora não interessa qual) e lança para a mesa (pilha de descarte) 1 carta a seu gosto; cada um dos outros jogadores, lança a seguir outra carta do mesmo naipe (caso o naipe esteja esgotado, terá de lançar uma carta de outro naipe); após terem jogado os 4, a vasa que fica na mesa será recolhida por aquele que lançou a carta mais alta, o qual a guarda ao seu lado até ao fim dessa volta; após 13 jogadas (quando já nenhum jogador tiver cartas na mão), contam-se as vasas que cada jogador arrecadou ao longo dessa volta (perde o que tiver mais).

Supondo que se pretende simular esse jogo por computador, diga justificando, como representava cada uma das 52 cartas do baralho e quais as estruturas de dados que usava para implementar as mãos, a pilha de descarte, e o conjunto de vasas junto por cada jogador.

Questão 5 (estruturas de dados, 4v)

Considere a estrutura de dados descrita pelos tipos abaixo declarados, que servem para suportar a Agenda de 1 mês num telemóvel Nokia:

```
typedef struct sTarefa
{
    int dia;
    int hora;
    char *tipo;
    char *descricao;
} Tarefa;

typedef struct sLLTarefa
{
    Tarefa t;
    struct sLLTarefa *seg;
} *LLTarefa, NodoLLTarefa;

typedef struct sAgendaMensal
{
    int mes;
    LLTarefa ag[31];
} AgendaMensal;
```

Com base nesta especificação desenvolva as seguintes alíneas:

- Codifique em C uma função `Tarefa leTarefa()` para ler os dados de uma tarefa (dia, hora, tipo, descrição) e criar uma célula com essa informação;
- Codifique em C uma função `AgendaMensal pushTarefa(AgendaMensal a, Tarefa t)` para inserir essa célula na cabeça da lista de tarefas desse dia;
- Codifique em C uma função `AgendaMensal sortedTarefas(AgendaMensal a, Tarefa t)` para inserir essa célula ordenadamente na lista de tarefas desse dia (ordem crescente de hora, sem repetições à mesma hora);
- Codifique em C uma função `void listarTarefas(AgendaMensal a, int dia)` para imprimir todas as tarefas agendadas para 1 dia (recebido como argumento).