

Programação Imperativa

1º Ano – LEI/LCC

Exame da 1ª Chamada – Duração: 2 horas

21 de Junho de 2007

Parte I

Esta parte do exame representa 12 valores da cotação total. Cada uma das 6 alíneas está cotada em 2 valores.

A obtenção de uma classificação abaixo de 8 valores nesta parte implica a reprovação no exame.

1. Considere a seguinte função que determina se existe algum elemento repetido nos primeiros n elementos de um *array* v . Caso haja duplicados retorna verdadeiro.

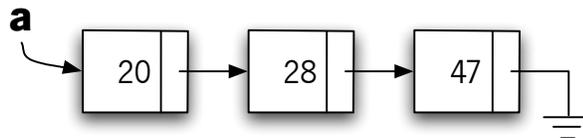
```
int dups (int v[], int n){
    int r = ...;
    int i, j;
    for (i = 0; ((i<n-1) && (r != 0)); i++)
        for (j=i+1; ((j<n) && (r != 0)); j++)
            r = v[i] - v[j];
    return (...);
}
```

- (a) Complete o código acima (inicialização de r e valor de retorno da função).
 - (b) Reescreva a função acima substituindo os ciclos `for(...)` por `while...`
 - (c) Suponha agora que essa função só será usada para *arrays* ordenados (por ordem crescente). Reescreva-a de forma a tomar partido desta restrição.
2. Considere as seguintes declarações para implementar listas ligadas de inteiros.

```
typedef struct slista {
    int valor;
    struct slista *seg;
} *Lista;
```

Lista a ;

- (a) Apresente o conjunto de instruções necessárias para que a variável a aponte para a lista desenhada abaixo.



- (b) Defina uma função `int ultimo (Lista)` que recebe uma lista **não vazia** e dá como resultado o último elemento.
- (c) Defina uma função `int ordenada (Lista)` que determina se uma dada lista ligada está ordenada por ordem crescente (em caso afirmativo a função retorna um valor diferente de zero).

Parte II

1. Relembre o jogo da forca do trabalho prático. Uma forma possível de armazenar as várias palavras agrupadas por temas é usar dois ficheiros: **Temas** e **Palavras**. No ficheiro **Temas** guarda-se um tema por linha. No **Palavras** guardam-se em cada linha as palavras correspondentes a esse tema, separadas por ponto e vírgula.
 - (a) Escreva uma função `void junta ()` que lê os ficheiros **Temas** e **Palavras** e produz um ficheiro **Enigmas** com uma linha por palavra, contendo essa palavra e o seu tema separados por uma vírgula.
 - (b) Escreva a função `void separa ()` com efeito inverso da anterior. Não deve assumir qualquer ordem no ficheiro **Enigmas**.
2. Maria e Hipólito 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).
 - (a) 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.
 - (b) Supondo que era a sua vez de se descartar, dada o conteúdo da pilha de descarte (1 a 3 cartas) e a sua mão actual, decida qual a próxima carta a lançar para a mesa.