

Programação Imperativa

1º Ano – LEI/LCC

Exame da 2ª Chamada – Duração: 2 horas

27 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 `moda`, que determina, num *array* de inteiros, qual o valor que ocorre com maior frequência.

```
int moda (int v [], int n){
    int m, mAux, f, fAux;
    int i, j;

    f = ...;
    i=0;
    while (i<n) {
        j=i+1; mAux = v[i]; fAux = 1;
        while (j<n) {
            if (v[j] == mAux) fAux++;
            j++;
        }
        if (fAux > f) { f = fAux; m = mAux; }
        i++;
    }
    return (...);
}
```

- (a) Complete o código acima (inicialização de `f` e valor de retorno da função); invoque a função `moda()` a partir da `main()`, passando como argumento um vector de comprimento 10, inicializado com 12,5,5,12,8,12,4,5,12,8.
 - (b) Reescreva a função substituindo o ciclo `while... interno` por `for(...)` e, se for possível, elimine a variável `mAux`.
 - (c) Suponha agora que a função só será usada para *arrays* ordenados (por ordem crescente). Reescreva-a de forma a tomar partido desta restrição.
2. Considere a seguinte função sobre listas ligadas de inteiros.

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

```
Lista f (Lista l) {
```

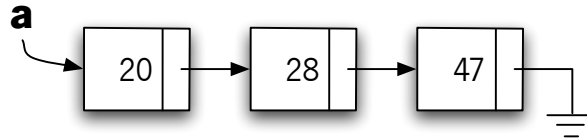
```

Lista r, aux;
r = NULL;

while (l != NULL) {
    aux = l ; l = l->seg;
    aux-> seg = r; r = aux;
}
return r;
}

```

- (a) Seja **a** uma variável do tipo **Lista** cujo valor é o correspondente ao desenho abaixo.



Indique, justificando a resposta, o valor de **a** após a instrução **a = f (a)**.

- (b) Defina uma função `float media(Lista)` que calcula média de uma lista **não vazia** de inteiros.
(c) Escreva as instruções necessárias para acrescentar o número 10 à cabeça da lista apontada por **a**.

Parte II

1. Considere a seguinte função para testar se uma dada matriz, quadrada de ordem **N**, é simétrica.

```

int eSimetrica (int a [N][N]) {
    int i, j;
    int r = 0;

    for (i=0;(i<N);i++)
        for (j=0;(j<N);j++)
            if (a[i][j] != a[j][i]) r = 1;
    return r;
}

```

Esta função é muito pouco eficiente porque: continua a testar depois de já saber que há um par que falha; testa todos os pares 2 vezes (compara **a[i]** com **a[j]** e ainda **a[j]** com **a[i]**). Modifique o código de forma a otimizar a função `eSimetrica()`, tomando em consideração os problemas apontados.

2. Escreva uma função `int allAlpha(FILE *f)` que lê todas as palavras de um ficheiro **f** passado como parâmetro e verifica se são todas constituídas apenas por letras. Para verificar cada palavra recorra a uma função auxiliar, implementada por si, equivalente à função do C (`isalpha()`).
3. Hipólito, cansado de se divertir com a **Arca de Jogos**, quer um programa para computador que o ajude a jogar à **Batalha Naval**, onde aliás ele perde sempre a Maria. Para isso precisa de representar o seu campo de batalha (**CB1**)—onde coloca a sua armada (4 submarinos de 1 cano, 3 corvetas de 2 canos, 2 fragatas de 3 canos, 1 porta-aviões de 4 canos) e onde aponta os tiros do adversário—e o campo de batalha do adversário (**CB2**)—onde regista os tiros que dispara contra o adversário e onde reconstitui a armada inimiga. Além disso, precisa de ir anotando os tiros que vai acertando em cada navio do adversário.
- (a) Com vista a poder implementar o referido programa, diga que estruturas de dados escolhia para armazenar os dados da **Batalha Naval**, descrevendo o esquema de representação que usava para guardar toda essa informação.
- (b) De acordo com o esquema da alínea anterior, desenvolva uma função `void jogada(tiro, tiro, tiro)`, que receba os 3 tiros da jogada adversária e indique o resultado; quanto caíram na água e quais os navios acertados.