

Programação Imperativa – EI (1º ano)

Mini-Teste 4A

Data: 30 de Abril de 2012

Hora: TP5

Dispõe de **40 minutos** para realizar este mini-teste.

Nome:

Número:

Questão 1 (produto cartesiano)

Foste encarregado de criar uma pequena aplicação para gerir os parques de estacionamento do Continente (aqueles inteligentes que têm uma luzinha a indicar se o lugar está livre ou ocupado). Considera que cada lugar num parque é identificado por um número inteiro. Um parque é definido pelo seu nome, o número de lugar que tem disponíveis e uma lista dos lugares que se encontram ocupados no momento. Quando um carro estaciona num determinado lugar é também registada a hora.

Considere os seguintes tipos de dados que foram especificados:

```
typedef char Nome[10];
```

```
typedef struct sHora
{
    int horas;
    int minutos;
} Hora;
```

```
typedef struct sLlugar
{
    int lugar;
    Hora h;
    struct sLlugar *seg;
} *Llugar, NLlugar;
```

```
typedef struct sParque
{
    int nlugares;
    Llugar ocupados;
    Nome nparque;
} Parque;
```

```
typedef struct sLParque
{
    Parque p;
    struct sLParque *seg;
} *LParque, NLParque;
```

Que poderão ser usados num programa da seguinte forma:

```
int main()
{
    Parque p1 = {600, NULL, "P1"},
             p2 = {1500, NULL, "P2"};
    LParque cp = NULL;
    ...
    cp = inserirParque(cp,p1);
    cp = inserirParque(cp,p2);
    ...
    listar(cp);
    ...
    if(disponivel( cp, "P1", 23 )
        cp = estaciona( cp, "P1", 23, consHora( 15, 20) );
    ...
    cp = saivatura( cp, "P2", 17 );
    ...
    listardisponibilidades(cp);
    ...
}
```

Especifique as funções utilizadas no exemplo:

void listar(LParque lp) - que lista no monitor todos os parques disponíveis juntamente com a sua informação: nome e número de lugares;

int disponivel(LParque lp, Nome p, int lugar) - que dá como resultado 0 (Falso) se o lugar *lugar* já estiver ocupado nesse parque e 1 (Verdadeiro) caso contrário;

LParque estaciona(LParque lp, Nome p, int lugar, Hora h) - que dá como resultado um novo sistema de parques após a entrada de um carro no parque *p*, que estacionou no lugar *lugar* à hora *h*;

LParque saivatura(LParque lp, Nome p, int lugar) - que dá como resultado um novo sistema de parques após a saída de um carro do parque *p*, do lugar *lugar*;

void listardisponibilidades(LParque lp) - que lista no monitor o nome de todos parques activos juntamente com o número de lugares disponíveis em cada um.