

# Programação Imperativa – EI (1º ano)

## Mini-Teste 4B

Data: 2 de Maio de 2012

Hora: TP3

Dispõe de <b>40 minutos</b> para realizar este mini-teste.
Nome: .....
Número: .....

### Questão 1 (produto cartesiano)

Foste encarregado de criar uma pequena aplicação para gerir uma agência matrimonial (aquelas onde uma pessoa à procura de um parceiro se dirige). O sistema de informação que irá suportar a aplicação é basicamente constituído por 3 entidades: a informação sobre a agência (nome, morada, telef, etc), a base de dados de pessoas registadas e uma lista de pedidos de "compatibilidade/matching". Foi-te pedido que desenvolvesse algumas funcionalidades sobre o modelo de dados do protótipo que se apresenta a seguir:

```
#define SIM 'S'
#define NAO 'N'

typedef struct sData
{
    int ano, mes, dia;
} Data;

typedef struct sAtributo
{
    char *designacao;    /* "fumador", "desportista", "música", ... */
    char valor;        /* SIM ou NAO */
} Atributo;

typedef struct sLAttrib /* a lista de atributos de uma pessoa */
{
    Atributo a;
    struct sLAttrib *seg;
} *LAttrib, NLAttrib;

typedef struct sPessoa /* a Pessoa */
{
    char* nome;
    Data nascimento;
    char sexo;
    char* cidade;
    LAttrib atributos;
} Pessoa;

typedef struct sLPessoa /* a Lista de Pessoas */
{
    Pessoa p;
    struct sLPessoa *seg;
} *LPessoa, NLPessoa;
```

```

typedef struct sPedido
{
    char* nome;    /* quem fez o pedido */
    char sexo;
    LAtrib atributos;
} Pedido;

typedef struct sLPedido    /* a Lista de Pedidos de Match */
{
    Pedido ped;
    struct sLPedido *seg;
} *LPedido, NLPedido;

typedef struct sAgencia
{
    char *nomeag;
    char *morada;
    char *telefon;
    LPessoa pessoas;
    LPedido pedidos;
} Agencia;

```

Que poderão ser usados num programa da seguinte forma:

```

int main()
{
    Agencia matrimonium;
    Pessoa p1 = {"Ana", {1982, 2, 7}, 'F', "Porto", NULL };
    LPessoa lpessoas = NULL;
    Pedido ped1;
    LAtrib carlosatrib = NULL;
    ...
    ped1 = consPedido( "Carlos", 'M', carlosatrib );
    ...
    p1 = InsAtrib( p1, "Fumador", NAO );
    ...
    matrimonium = InsereCandidato( matrimonium, p1 );
    ...
    ListarFumadores( matrimonium );
    ...
    lpessoas = Match( matrimonium, 2 );
    ...
}

```

Especifique as funções utilizadas no exemplo:

**Pedido consPedido( char\* n, char s, LAtrib la )** - que constrói um *Pedido* com os valores que lhe são passados;

**Pessoa InsAtrib( Pessoa p, char\* id, char v )** - que acrescenta o atributo com nome *id* e valor *v* à lista de atributos da pessoa *p* e devolve a nova pessoa;

**Agencia InsereCandidato( Agencia a, Pessoa p )** - que dá como resultado uma nova agência resultante da inserção na nova pessoa;

**void ListarFumadores( Agencia a )** - que lista no monitor as pessoas (nome, data de nascimento e sexo) que têm "Fumador" com o valor SIM na sua lista de atributos (a ausência do atributo pode ser assumida como NAO);

**LPessoa Match( Agencia a, int npedido )** - que dá como resultado a lista de pessoas que têm um *match* superior a 50% com a pessoa que consta do pedido na posição *npedido* da lista de pedidos. O *match* é calculado pelo quociente entre o número de atributos em comum e o total de atributos de cada pessoa registada na agência;