

## Tarefa Consulta

Uma clínica médica utiliza um sistema informatizado para auxiliar no gerenciamento de informações sobre pacientes e suas consultas. Nesse sistema, as informações referentes as consultas são armazenadas em um vetor de ponteiros para o TAD *Consulta*.

Quando um médico não pode comparecer à clínica em um determinado dia, ele é substituído por outro profissional no atendimento aos seus pacientes. USANDO o TAD *Consulta*, escreva uma função que efetua a substituição de um médico responsável pela consulta de pacientes agendados para uma determinada data.

A função recebe como parâmetros um vetor de ponteiros para consulta (*vet*) e o tamanho desse vetor (*n*), a data da substituição (*d*), o nome do médico que precisa faltar (*med*) e o nome do substituto (*sub*).

A função retorna um inteiro indicando o número de substituições efetuadas. Considere que os elementos do vetor recebido como parâmetro estão em ordem aleatória.

O protótipo da função é o seguinte:

```
int troca_medico(Consulta **vet, int n, Data d, char *med, char *sub);
```

onde *Data* corresponde ao tipo estruturado descrito abaixo.

```
struct data {
    int dia, mes, ano;
};
typedef struct data Data;
```

Inclua as definições de estrutura exatamente como acima no seu arquivo de resposta, mas não inclua a função main. Inclua também a referência a quaisquer arquivos de biblioteca que você use (e.g., stdio.h).

O TAD *Consulta* tem a seguinte interface (arquivo *Consulta.h*):

```
typedef struct consulta Consulta;
```

<pre>Consulta *Cons_cria_consulta (char *paciente, int d, int m, int a, int hr, int, min, char *medico);</pre>	Aloca dinamicamente área para o tipo <i>Consulta</i> , inicializa essa área com os valores recebidos como parâmetros e retorna o ponteiro para essa área (ou NULL se não foi possível alocá-la).
<pre>Consulta *Cons_copia_consulta (Consulta *p);</pre>	Copia uma consulta, ou seja, aloca uma nova área dinamicamente, inicializa essa área com valores idênticos ao da consulta recebida como parâmetro e retorna o ponteiro para a nova área (ou NULL se não foi possível alocá-la).
<pre>void Cons_altera_paciente (Consulta *p, char *nome);</pre>	Altera o paciente a ser atendido em uma consulta. O ponteiro para a consulta e nome do novo paciente são recebidos como parâmetros.
<pre>void Cons_altera_data_consulta (Consulta *p, int d, int m, int a);</pre>	Altera a data de uma consulta. O ponteiro para a consulta e os novos valores de dia, mês e ano são recebidos como parâmetros.
<pre>void Cons_altera_medico (Consulta *p, char *med);</pre>	Altera o médico responsável por uma consulta. O ponteiro para a consulta e o nome do médico substituto são recebidos como parâmetros.
<pre>int Cons_verifica_data_consulta (Consulta *p, int d, int m, int a);</pre>	Compara uma data, fornecida como parâmetro, com a data de uma consulta. O ponteiro para a consulta e o dia, mês e ano a serem comparados são recebidos como parâmetros. Retorna 1, se as datas são iguais, ou 0, em caso contrário.
<pre>int Cons_verifica_medico (Consulta *p, char *med);</pre>	Compara um médico, recebido como parâmetro, com o médico responsável por uma consulta. O ponteiro para a consulta e o nome do médico a ser comparado são recebidos como parâmetros. Retorna 1, se os nomes dos médicos são iguais, ou 0, em caso contrário.
<pre>void Cons_libera_consulta (Consulta *p);</pre>	Libera a área de uma consulta cujo endereço é recebido como parâmetro.