

Considere que uma imobiliária mantém os imóveis para venda em um vetor de estruturas (Fig. 1a), onde cada estrutura representa um imóvel (Fig. 1b). Essa estrutura é composta por 3 elementos: uma estrutura que representa o endereço do imóvel (i.e., uma estrutura dentro da outra), o número de quartos e o valor em R\$. O endereço, por sua vez, é uma estrutura contendo a rua (*string*), o número do prédio (*int*), o complemento (*string*) e o bairro (*string*). Considere que as *strings* não têm mais do que 10 caracteres.

[i] Escreva uma função que cria o vetor de estruturas a partir do seguinte arquivo texto, onde os nomes representando *strings* têm um único termo (e.g., o nome de rua VPat ao invés de Voluntários da Pátria e Apt112 ao invés de Apt 112) e a primeira linha contém o número de linhas a serem lidas:

```
6
VPat 12 Apt112 BOTAFOGO 3 quartos 700000
BMit 77 Apt734 LEBLON 2 quartos 800000
BRib 85 Apt914 COPACABANA 1 quarto 700000
SClem 55 Apt401 BOTAFOGO 2 quartos 700000
RGrand 7 Apt503 BOTAFOGO 3 quartos 650000
NSCopa 79 Apt202 COPACABANA 1 quarto 600000
```

vetor

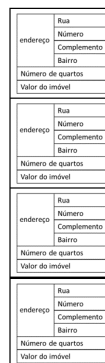


Fig. 1a

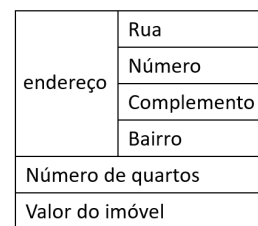


Fig. 1b

Note que a palavra “quarto” pode estar no singular ou no plural. Essa função deve retornar um vetor alocado dinamicamente e também deve disponibilizar, via argumento tipo ponteiro, o número de registros do arquivo (6, no exemplo acima).

[ii] Escreva uma função que imprime apenas um imóvel, dado o ponteiro para a estrutura que o representa. Escreva também uma outra função que imprime todo o vetor de estruturas, usando a função que imprime um imóvel.

[iii] Escreva uma função que:

- recebe um vetor de imóveis, o número de imóveis, um bairro e um número de quartos. Essa função deve retornar, via argumento, o número de imóveis que atendam simultaneamente aos critérios bairro e número de quartos;
- cria e retorna um novo vetor de PONTEIROS para imóveis, alocado dinamicamente do tamanho exato necessário, com os ponteiros para os imóveis do vetor original recebido que atendam simultaneamente aos critérios bairro e número de quartos (Fig. 2). Note que apenas o vetor deve ser alocado dinamicamente; os elementos apontam para estruturas já existentes. Não esqueça que também deve ser devolvido, em uma variável inteira cujo endereço deve ser fornecido na chamada da função, o número de imóveis que atendam ao critério.

Obs: Caso não exista nenhum imóvel que atenda aos critérios, nenhum vetor é criado e deve ser retornado NULL. Caso não seja possível criar o novo vetor, também deve ser retornado NULL.

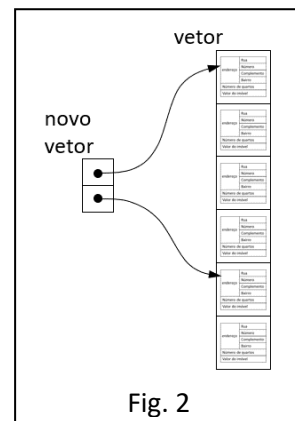


Fig. 2

No exemplo acima, se o critério é bairro BOTAFOGO e número de quartos 3, o novo vetor aponta para as estruturas que estão na primeira posição e na penúltima posição. Neste caso, o número de imóveis identificados é 2. Para testar esta função faça um função auxiliar que imprime o vetor de ponteiros (e use obrigatoriamente a função que imprime um único imóvel do item [ii]).

[iv] Escreva uma função que gera um código (*string*) a partir do nome e da data de nascimento do cliente da imobiliária (para manter o sigilo). A imobiliária faz isto intercalando os caracteres do primeiro nome do cliente com os 6 (seis) caracteres que representam a data de nascimento deste cliente. A *string* gerada é sempre completada com os caracteres restantes do nome ou da data. Por exemplo, nome = “ANA” e data = “261281” geram a cadeia “A2N6A1281”, enquanto que nome = “CRISTINA” e data = “261281” geram “C2R6I1S2T8I1NA”. Esta função deve receber o nome e a data e retornar uma nova string alocada dinamicamente e do tamanho exato necessário. Se não houver memória suficiente, a função deve retornar NULL.

A Fig. 3 ilustra o *output* dos testes para os exemplos de todos os itens acima. Imprima o código de um cliente, escolhendo livremente exemplos de nomes e datas de nascimento, junto com as ofertas encontradas com o uso da função do item [iii].

```
Vetor de Imoveis:
VPat 12 Apt112 BOTAFOGO 3 quarto(s) R$ 700000.0
BMit 77 Apt734 LEBLON 2 quarto(s) R$ 800000.0
BRib 85 Apt914 COPACABANA 1 quarto(s) R$ 700000.0
SClem 55 Apt401 BOTAFOGO 2 quarto(s) R$ 700000.0
RGrand 7 Apt503 BOTAFOGO 3 quarto(s) R$ 650000.0
NSCopa 79 Apt202 COPACABANA 1 quarto(s) R$ 600000.0
2 ofertas para C2R6I1S2T8I1NA, COPACABANA com 1 quarto:
BRib 85 Apt914 COPACABANA 1 quarto(s) R$ 700000.0
NSCopa 79 Apt202 COPACABANA 1 quarto(s) R$ 600000.0
2 ofertas para A2N6A1281, BOTAFOGO com 3 quartos:
VPat 12 Apt112 BOTAFOGO 3 quarto(s) R$ 700000.0
RGrand 7 Apt503 BOTAFOGO 3 quarto(s) R$ 650000.0
```

Fig. 3

Gerencie problemas de memória e feche o arquivo adequadamente. No caso de insuficiência de memória, apenas a *main* deve interromper a execução. No caso de inexistência de arquivo para leitura, você pode interromper a execução em qualquer local e função que você queira. **ATENÇÃO:** use dados completamente diferentes dos exemplos acima.