

1. Use a classe abaixo, sem atributos, e sem construtor, para abrigar os métodos seguintes.

```
public class Lista11 {  
  
    /** a) Um método recebe como parâmetro dois arrays de inteiros  
    /*      de nomes a e b, exibe-os na tela, e retorna a  
    /*      quantidade de elementos de a que aparece em b. */  
  
    /** b) Programe um método que recebe um array de nomes de  
    /*      pessoas e retorna um array de char onde cada posição  
    /*      contém a inicial do nome da posição correspondente do  
    /*      array de nomes.      */  
}
```

Crie uma classe de testes chamada **Lista11Teste** para testar os métodos criados na classe anterior.



2. Na venda de passagens para as viagens de ônibus, uma empresa possui uma classe **Viagem** com os seguintes atributos privados: código da viagem (int) e um array de Strings indicando o nome do passageiro que ocupa cada lugar do ônibus. Se o ônibus tem 23 lugares, serão numerados de 1 a 23. Assim, se a passageira “Nara Leão” comprou o lugar de número 12, seu nome será colocado na posição 12 do array.

Crie a classe **Viagem**, com dois construtores e os seguintes métodos.

Construtor 1: tem dois parâmetros: o código da viagem e a quantidade de lugares do ônibus. Instanciar o array de lugares, simulando o ônibus ainda vazio.

Construtor 2: será sem parâmetros: o código da viagem deverá ser lido do teclado e a quantidade de lugares do ônibus deverá ser 12 ou 23 ou 41, escolher uma dessas opções usando algum tipo de sorteio com randômicos.

+ *listaLugares* — exibe o número de cada lugar e o nome do seu ocupante, ou a palavra “livre”, se ele ainda não foi vendido

+ *listaLugaresDisponíveis* – mostra na tela os números apenas dos lugares vagos.

+ *vendeLugar* — recebe o nome do passageiro que deseja adquirir um lugar. Se o ônibus já está lotado, deve ser exibida esta informação e terminar o método. Exibir na tela a lista dos números dos lugares disponíveis (chamar o método anterior) para que o passageiro faça sua escolha (ler do teclado o número escolhido pelo passageiro, com validação). A validação deve ser com repetição, até que o usuário digite um lugar válido e disponível. O método retorna `true` ou `false`, de acordo com o sucesso ou fracasso na venda de lugar.

- *buscaPassageiro* – recebe um nome de pessoa e verifica se ela é passageira no ônibus, retornando o número do lugar que ela ocupa ou 0, se ela não está na viagem.

+ *desistencia* — recebe o nome de um passageiro e procura por ele no array (chamar o método anterior para isso). Se encontrar, retira-o, liberando o respectivo lugar e retorna `true`. Se não encontrar, retorna `false` para indicar o insucesso da busca.

+ *gets*

3. Faça uma classe de teste que simule o movimento de venda de passagens da rodoviária, para uma **Viagem** (conf. ex. 2 da lista). Inicialmente, instancie uma viagem. A seguir faça um menu repetitivo com as opções *Vende Lugar*, *Desistência*, *Lista lugares* e *Encerra*. Conforme a opção escolhida pelo usuário, realize a operação correspondente. Use mensagens na tela para facilitar o acompanhamento (por ex: *Venda completada*, *Passageiro não encontrado*, etc...). No caso de opção inválida, exiba a mensagem *Redigite!*

