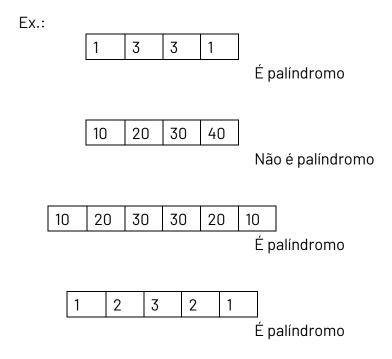
PILHA - Exercícios

Estrutura de Dados e Armazenamento

1- Utilizando a classe Pilha feita em sala, implemente o método ehPalindromo, na classe onde está o main. Esse método recebe um vetor de inteiros e retorna true se a sequência de inteiros no vetor for palíndroma e retorna false, caso contrário.

No main, inicialize um vetor de inteiros com valores e teste se esse vetor é ou não palíndromo.



2- Conversão de decimal para binário. Utilize uma pilha para converter um número de decimal para binário.

Ex: Número 9 - representação em binário: 1001

Número 12 – representação em binário: 1100

Número 32 - representação em binário: 100000

Dica: para converter um número de decimal para binário, devemos dividi-lo sucessivamente por 2, até que o quociente seja zero. A representação em binário corresponde aos restos de cada divisão, considerados na ordem inversa.

Assim, basta ir dividindo o número sucessivamente por 2, e ir empilhando os restos, até que o quociente seja zero. Depois, ir desempilhando e imprimindo o valor desempilhado.

3- Criar a classe PilhaObj, semelhante à classe Pilha, porém ao invés de armazenar int, deverá armazenar elementos de tipo genérico <T>, como foi feito em ListaObj. Faça as alterações necessárias para que a classe PilhaObj funcione adequadamente.



No main, teste sua classe PilhaObj:

- crie um objeto da classe PilhaObj para armazenar Strings
- empilhe algumas Strings
- exiba a pilha
- desempilhe as Strings e exiba-as conforme desempilhar
- 4- Implementar um programa que utilize uma pilha para ler uma frase e exibi-la invertida. Neste caso, o elemento da pilha deve ser do tipo Character (use a PilhaObj).

Ex: a frase "A pilha do gato" deve sair "otag od ahlip A"

Dica Java: para trabalhar com cada caractere da String, pode-se converter a String num array de char's, utilizando o método toCharArray().

```
Ex: String frase= "A pilha do gato";
   Character[ ] fraseChar= frase.toCharArray( );
   for (int i = 0; i < fraseChar.length; i++) {
        System.out.println(fraseChar[i]);
   }</pre>
```

Ou pode-se obter cada caractere da String, utilizando o método charAt(índice)

```
Ex: for(int i=0; i < frase.length(); i++) {
         System.out.println(frase.charAt(i));
}</pre>
```

5- Implementar um programa que utilize uma pilha para verificar se uma dada cadeia de caracteres é ou não palíndroma. Uma cadeia de caracteres é palíndroma quando tem a mesma leitura, lendo-se da esquerda para direita ou da direita para esquerda (sem considerar os espaços em branco, tracinho, acento, letra maiúscula/minúscula).

Por exemplo: radar, ovo, Ana, subi no ônibus, Socorram-me subi no ônibus em Marrocos, anotaram a data da maratona, a torre da derrota

6- Crie um projeto para este exercício.

Nesse projeto, crie uma classe chamada Livro, com atributos nome e autor. Implemente os getters e setters, o construtor e o toString().

Copie a classe PilhaObj para este projeto. Implemente os métodos a seguir na classe PilhaObj:



```
PilhaObj <T> multiPop (int n)

// esse método desempilha n elementos da pilha, empilhando-os numa pilha

// auxiliar (um de cada vez), e retorna essa pilha auxiliar

// se n é maior do que o número de elementos da pilha, retorna null

// senão cria uma pilha aux (objeto da classe PilhaObj <T>) e empilha em //
aux os n elementos desempilhados da pilha

void multiPush (PilhaObj <T> aux)

// esse método recebe uma pilha aux como elementos empilhados

// os elementos da pilha aux deverão ser empilhados na pilha, um a um
```

Nesse mesmo projeto, crie uma classe chamada Teste, e configure para que ela tenha o método main

No main:

- crie um objeto da classe PilhaObj <Livro>, chamado pilha
- crie vários objetos da classe Livro e empilhe-os, um a um, na pilha.
- exiba a pilha
- Simule a situação em que você deseja tirar um livro da pilha, mas ele não está no topo da pilha. Então, você deve usar os métodos acima multiPop e multiPush, para "tirar" o livro desejado da pilha, mantendo os demais livros empilhados como inicialmente.

