ITCS 208 Object Oriented Programming	Name:	Lab Score	Challenge Bonus	Peer Bonus
	ID:			
	Group:			

Lab05: Array and ArrayList

You are hired to develop a song application to manage a song playlist. In this application, you have to create two classes named **Song** and **`Playlist`**. The `main` method is already provided in `SongApp.java`

Task1: Create a **Song** class and implement the following attributes and methods:

- Attributes:
  - A song's title (String), and a song's duration in minute (double)
- Constructor method:
  - `public Song(String title, double duration)`
- Getter methods:
  - `getTitle()` // return a song's title in **String** data type
  - `getDuration()` // return a song's duration in minutes in **double** data type
  - `getDurationInSec()` // return a song's duration in seconds in **int** data type
  - `toString()` // return a song's information in **String** data type
    // The String result must be in the following format

{title}: *{duration}* minutes (*{duration}* seconds) e.g., `Perfect: 4.21 minutes (261 seconds)`

Task2: Create a **`Playlist`** class and implement the following attributes and methods:

- Attributes:
  - A playlist's name (String), a list of songs (using ArrayList to store Song objects), and total duration of song in seconds (int)
- Constructor method:
  - `public Playlist (String name)`
- Other methods:
  - `void addSong(Song song)`
    - to add a new song into the playlist
  - `void addSongAtIndex(Song song, int index)`
    - to add a new song into the playlist at a given position. If the index is invalid, you must display an error message (see expected output). Note that the index in the playlist starts at '0'.
  - `boolean removeSongByIndex(int index)`
    - to remove an existing song from the playlist at a given index
    - return true if there was a song at the given index, otherwise display an error message and return false
  - `boolean removeSongByTitle(String title)`
    - to remove an existing song from the playlist by looking at the song title
    - return true if the song exists, otherwise show an error message and return false
  - `void moveUp(int position)`
    - to rearrange the position of songs in the playlist by moving the song at the given index up by one position.
  - `void moveDown(int position)`
    - to rearrange the position of songs in the playlist by moving the song at the given index down by one position.
  - `void showPlaylist()`
    - to print the playlist on the console in the following format

playlist name
[0] *title*, *duration* minutes (*duration* seconds)
[1] *title*, *duration* minutes (*duration* seconds)
….
Total duration is *xx.xx* minutes

Note that: For add and remove methods, you suppose to show (print out) an error message when the given index or title does not existing in your current playlist. Pleas see the expected output for some example messages. Do not forget to update the total duration of the playlist when you add/remove songs as well. **Each minute only have 60 seconds. So 3.30 minute + 3.40 minutes = 7.10 minutes (not 6.70 minutes)**

Here is the expected output after you run the SongApp.java class.

```
Welcome to SongAPP

Add songs -------------------------
My Favorite Songs Playlist
[0] End Game, 4.11 minutes (251 seconds)
[1] Perfect, 4.21 minutes (261 seconds)
[2] Anywhere, 3.35 minutes (215 seconds)
[3] How long, 3.3 minutes (210 seconds)
Total duration is 15.37 minutes

Rearrange songs --------------------
My Favorite Songs Playlist
[0] Perfect, 4.21 minutes (261 seconds)
[1] End Game, 4.11 minutes (251 seconds)
[2] How long, 3.3 minutes (210 seconds)
[3] Anywhere, 3.35 minutes (215 seconds)
Total duration is 15.37 minutes

Remove songs ----------------------
My Favorite Songs Playlist
[0] Perfect, 4.21 minutes (261 seconds)
[1] How long, 3.3 minutes (210 seconds)
Total duration is 7.51 minutes

Check error -----------------------
Error: Couldn't add song at index 3
Error: The title is not found
Error: The index is invalid
```

Array Practice (BONUS)

Write a Java program to check palindrome. A palindrome is a word, phrase, number, or other sequence of characters which can be read the same way from backward or forward. For example, “madam”, “mom”, “abcba”, and “123321” are single word palindrome. “Don’t nod.”, “Top spot” and “No lemon, no melon” are multiple words palindrome. *(Punctuation and spaces between the words or letter is allowed.)* “Java”, and “Array” are NOT palindrome. The longest palindrome in Oxford dictionary is “tattarrattat”.

a. You may write a `Class` or simply have everything in the `main(String[] args)` method.
b. Your program has to accept input `String` from users and check whether the input `String` is a palindrome or not. An example output is shown in the box below:

```
Enter a word or phrase to check if it is a palindrome: tattarrattat
The input word “tattarrattat” is a palindrome

Enter a word or phrase to check if it is a palindrome: I love java
The input phrase “I love Java” is not a palindrome
```

Peer Bonus (Optional):