# Programming Exercise 15.3

**MP3 Shuffler, v.2.0**

**Purpose**. Learn to use "collections".

This improves the MP3 player's shuffle feature to make sure that a recently played song is not selected again so soon after it was last played.

**Requirements**. Write a new version of Exercise 13.3's `mp3Shuffler1.py`, named **`mp3Shuffler2.py`**. Use a collection to remember the last 5 songs played. If the program selects a song that was played in the last 5, choose again. Keep choosing until a song that has *not* been played in the last 5 is found and "played".

Use a collection to store the last 5 played songs. Add the last-played song to the *front* of the list, and if the list size is then greater than 5, use `song.pop()` to remove the oldest song from the list -- assuming the list is named `song`.

This requires a validation loop after an index for the next song is randomly generated. Exit the validation loop only if the song is not in the collection.

**Program I/O.** <u>Input</u>: user enters Y to "play" a song, or N to exit. <u>Output</u>: a song title console screen, in response to every Y from the user.

**Example.** Your program's console I/O should look something like this, with user input in blue:

```
play a song [Y/N]?: y
Hey Jude

play a song [Y/N]?: y
Imagine

play a song [Y/N]?: y
Johnny B. Goode

play a song [Y/N]?: y
Respect

play a song [Y/N]?: y
Good Vibrations

play a song [Y/N]?: y
Smells Like Teen Spirit

play a song [Y/N]?: y
Hey Jude

play a song [Y/N]?: n
```