Last time you developed a *Song* class for the BobCatPod, the Texas State version of the i-pod.

Now it’s time to develop the BobCatPod itself, which will handle more than one song. You need to create a class *BobCatPod* that contains the following data:

- The memory size (int) of the BobCatPod in MB.
  The number of MB of memory will be set when the tsuPod is "constructed" and cannot exceed 256MB.
- Number of songs (int)
  The number of songs must be maintained as songs are added and deleted from the list.
- A list of the songs.
  The song list should be implemented as a linked list.

```c
struct SongNode
{
    Song s;
    SongNode *next;
};

SongNode *songs; //the head pointer
```

*BobCatPod* contains the following functions and operations:

- **int addSong(Song s)**
  inserts s at the front of the list and returns 1 if successful, 0 if not (memory exceeded)
- **int removeSong(Song s)**
  deletes s from the list and returns 1 if successful, 0 if not (song s not found)
- **void showSongList()**
  exploits overloaded << operator
- **int getTotalMemory()**
  returns total memory
- **int getRemainingMemory()**
  item returns total unused memory

In addition, add operators to the *Song* class as follows:

- overloaded == operator
  This can be used to implement removeSong
- overloaded < and > operators
  For < and > use artist, then title, then size (lexicographic ordering). These will be used later.

Use the same style of file organization (.h and .cpp files) as for the Song class. Document all code. Your code must run with the driver supplied on the course website.