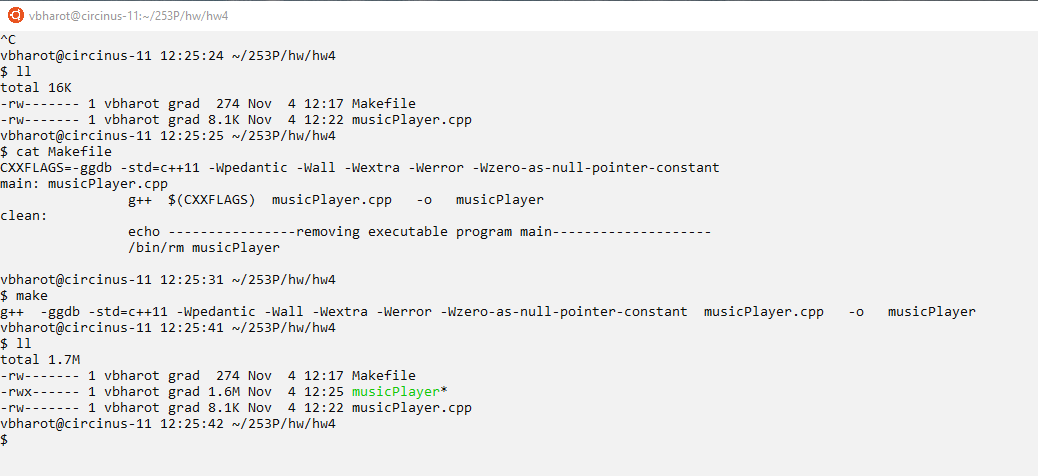
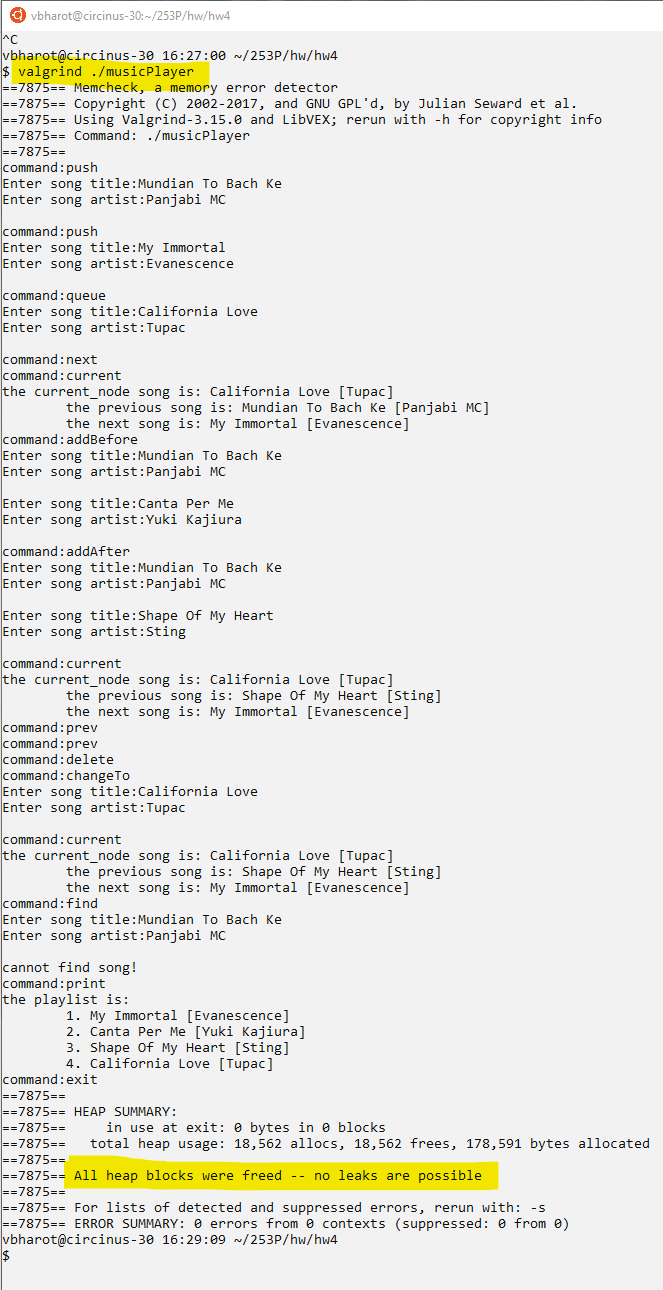
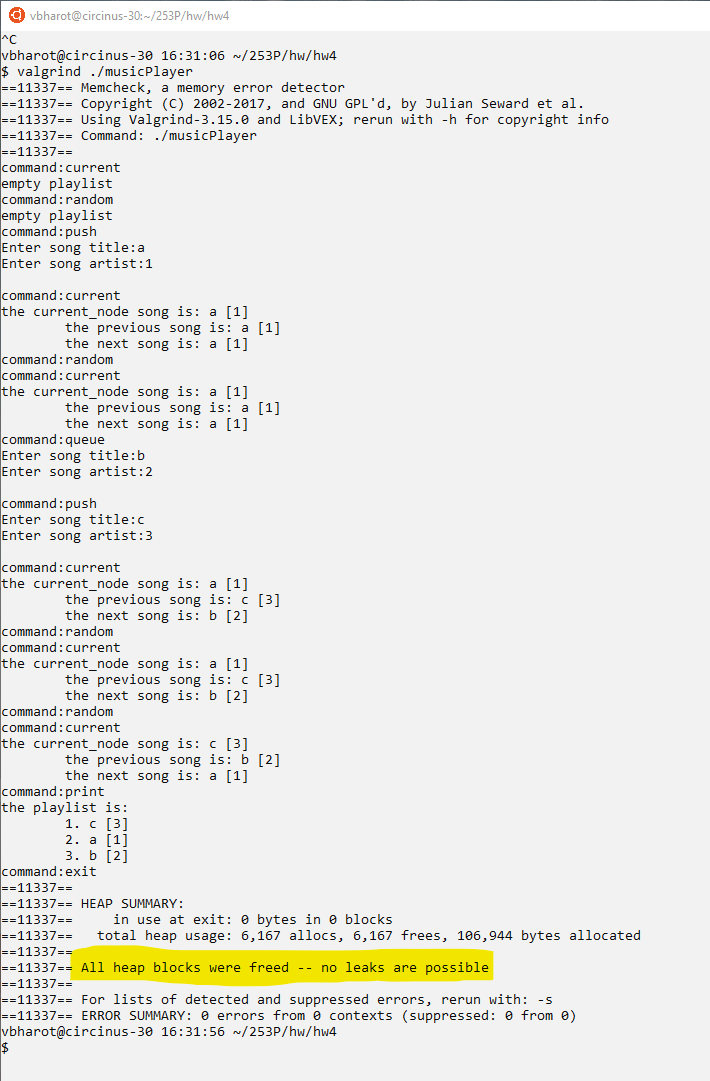
Compilation:



Example from assignment:



Provided test input:



1. What data structure did you implement ***SimplePlayList*** as?
   1. Singly linked list.
2. List all the attributes (aka fields) you needed in order to implement ***SimplePlayList*** (also include the attributes for any other auxiliary data structures it uses)?  (Do not list functions (aka methods)).
   1. SimplePlaylist
      * Current\_node pointer
      * Head node pointer

Where each Node is

Song Node

* + - Next Node pointer
    - Title string
    - Artist string

1. How does ***SimplePlayList*** retrieve a random song in *O(n)* time?  Explain in detail using a few sentences.
   1. First get the total number of songs present in the lined list in O(n).
   2. Then choose a random number from 1 to number of song O(1).
   3. Then iterate the current pointer from the head, random number of times obtained from step 2. and return the final resting current pointer at random position.
2. If ***prev*** is processed in *O(n)* time, then how is ***find*** able to print the previous song of the found song in *O(1)* time?
   1. While finding the required song we will maintain another pointer called previous pointer which is always behind the current pointer which is used to find the required song. And then we will use both the current pointer and previous pointer to print required songs.
   2. As we will only do one pass for find it’s O(n) which will give us both the current required and previous pointer.

148 sortList:



1019 Next Greater Node in Linked List:

