

A4 Grading Guide

Step 1: Comment Headers (1 Point)

First you should check to make sure that the students have correctly updated the header comments for a4.py.

Step 2: Style Points (2 Points)

Check over student code to make sure they have removed pass statements and have no lines over 80 characters. They should also have removed any debugging print statements for the submission on CMS. Take one point (max of 2) for each error in this regard.

Step 3: Run Autograder

The next step is to run the autograder on students' work. This will help check correct functionality and make future steps quicker.

Step 4: songs_in (27 points)

Control Flow Logic (7 points)

- 3 points: songs_in is called recursively on Mix objects only.
- 3 points: there is a for loop that iterates over the contents of the Mix (a correct solution that manages to not use a for loop is also acceptable)
- 1 points: final list is sorted

Functionality (20 points)

A submission that passes the autograder earns all 20 points. Otherwise, partial credit can be awarded as follows:

- 2 points: base case: a Mix with one song
- 2 points: base case: a Mix with multiple songs
- 4 points: recursive case: a Mix with one song and a submix (no duplicate songs)
- 4 points: recursive case: a Mix with one song and a submix (duplicate songs)
- 4 points: recursive case: a Mix with multiple mix (no duplicate songs)
- 4 points: recursive case: a Mix with multiple mix (duplicate songs)

Step 5: basic_mixes (32 points)

Control Flow Logic (10 points)

- 3 points: basic_mixes is called recursively on Mixes objects only
- 3 points: there is a for loop that iterates over the contents (a correct solution that manages to not use a for loop is also acceptable)
- 3 points: checks that `m` is basic aka checks that `m` contains only songs

- 1 points: final list is sorted

Functionality (22 points)

A submission that passes the autograder earns all 22 points. Otherwise, partial credit can be awarded as follows, **except: the autograder will double-count forgetting to allow repeats.**

Students should only lose 4 points, not 8, if they mistakenly remove duplicates.

- 2 points: base case: a Mix with no reachable submix
- 4 points: recursive case: a Mix with only one reachable submix
- 4 points: recursive case: a Mix with multiple song-only submixes (containing duplicates)
- 4 points: recursive case: a Mix with multiple song-only submixes containing no duplicates
- 4 points: recursive case: a Mix with deeply nested submixes containing duplicates
- 4 points: recursive case: a Mix with deeply nested submixes containing no duplicates

Step 6: mixes_with (38 points)

Control Flow Logic (18 points)

- 3 points: there is a for loop over mlist
- 3 points: there is a for loop over the contents of each mlist element
- 3 points: case analysis for if the contents of the mlist element is equal to the song or it is a Mix
- 3 points: mixes_with is recursively called on Mixes within the contents of each mlist element
- 3 points: the title of the mix is correctly added to the accumulator list
- 3 points: the resulting list has no duplicates and is sorted

Functionality (20 points)

A submission that passes the autograder earns all 20 points. Otherwise, partial credit can be awarded as follows:

- 2 points: base case: the song isn't found in mlist
- 2 points: base case: song is directly inside the mix (first layer), mlist has one element
- 2 points: base case: song is directly inside the mix (first layer), mlist has more than one element
- 2 points: base case: song is directly inside multiple elements of mlist
- 4 points: recursive case: song is indirectly inside a mix (second layer), mlist has one element
- 4 points: recursive case: song is indirectly inside a mix (second layer), mlist has more than one element
- 4 points: recursive case: song is indirectly inside multiple elements of mlist
- **Exception:** deduct two points if the code cannot handle the case where the Song is contained in a non-basic mix.

TOTAL OUT OF 100 POINTS