

Computer Programming 2 Test 3

09/05/2022

Lasse Loepfe

- You have 2 hours to complete the assignment.
- If the code does not compile, the exercise won't be accepted for submission.
- Tests are provided to check if your code is correct. You cannot modify the tests!
 - If a test does not pass, that exercise will get at most half of the points.
- Code is expected to be readable, clean, and optimal.
- A skeleton of the exercise is provided. Feel free to add more files and/or include more libraries of your need them.
- To check if your code is correct, you can write the code you need in the `main()` of the file `main.cpp`, but leave it unchanged before submitting the exam.
- Inside the code, replace "TYPE YOUR NAME HERE" with your complete name.
- When you finish, ZIP the whole folder with a filename called "lastname_name.zip" and upload it to the "Test3" folder.

Exercise 1: (4 points)

Implement the method `maxValue(BSTNode* root)` so that it returns the highest value stored in a given Binary Search Tree as efficient as possible.

Exercise 2: (6 points)

You are in charge of implementing some functionalities in the famous game Super Mario Kart. More precisely, you have to write a couple of methods that allow consulting and modifying the positions of some players during a race, as these should change each time a player advances another one.

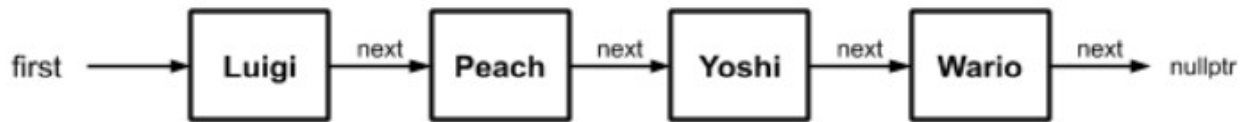
The container used to represent ranking containing the players sequentially is a singly-linked list. The ranking contains a pointer to the first player node, and each node contains the name of the player, and a pointer to the next player node. The order of nodes in the ranking reflects the position of the players in the race.

You have to implement the following two methods:

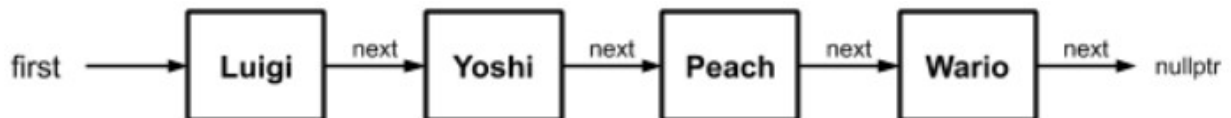
- (3 points) `int find_player_position(const char *name) const`
It has to traverse the list of nodes and return the position of the player whose name is the same as the one passed as parameter. If the player is not found in the list, -1 must be returned.

- (3 points) void advance(int player_position)
It has to move the player at the specified position one position forward.

Let's see some examples to clarify: If we have the following Ranking:



- If we invoke find_player_position("Luigi"), the returned value will be 0.
- If we invoke find_player_position("Yoshi"), the returned value will be 2.
- If we invoke find_player_position("Mario"), the returned value will be -1 (because "Mario" cannot be found in the list).
- If we invoke advance(0), the list is unchanged (because the first player "Luigi" is already at the first position).
- If we invoke advance(2), the player at position 2 "Yoshi" will advance "Peach", and the resulting list will be like this.



NOTE: You are not allowed to change the name member of any node. In order to move nodes back and forth in the list, you have to update the next pointer of all nodes involved in