Midterm exam

Overview

You are to complete some code for a hypothetical Role Playing Game (RPG), specifically code for having a party of characters travel over different types of terrain. You have been given `main.cc` which includes test cases. You have also been given `rpg.h`, which includes some function prototypes, but is missing some data type definitions. In addition to completing `rpg.h`, you will need to write, from scratch, `rpg.cc` (or `rpg.cpp`), which will implement the needed functions. You have also been given a Makefile, which you may use during testing, but it is not required.

In short, you will need to submit two files: `map.h` and `map.cc`. Please zip them up together into a .zip file before submitting.

Requirements

Note that you may get partial marks for any part, in the case that it is not complete or not totally correct.

1. Modify `rpg.h` to include an enumeration class called `terrain_type` with 3 terrain types: `water`, `sand` and `volcano` (2 marks)
2. Modify `rpg.h` to include a struct called `character` which gives the attributes `walking_speed` and `flying_speed`, both ints (2 marks)
3. Add an implementation of `int travel_speed(character const* c, terrain_type t)` to `rpg.cc`. If the terrain type is `water`, then the flying speed should be returned, since walking is impossible. If the terrain type is volcano, then the walking speed should be returned, since flying over a volcano is impossible (or at least in this RPG it is). If the terrain type is sand, then flying and walking are both possible, so you should return whichever is largest. However, since sand is difficult to walk on, you must use a walking speed of (walking_speed+1)/2. (5 marks)
4. After finishing step 3, the `CHECK_TS1` test cases in `main.cc` should pass.
5. Add an implementation of `int travel_speed(character const* cs, size_t n, terrain_type t)` to `rpg.cc`, which considers the travel speed of an entire group of characters, not just a single character. The `n` parameter indicates the number of characters in the array. You must return the *minimum* travel speed of all characters in the group, given the terrain type, because the group travels at the speed of its slowest member. (4 marks)
6. After finishing step 5, the `CHECK_TS2` test cases should pass.
7. Add an implementation of `int travel_speed(character const* cs, size_t cs_n, terrain_type const* ts, size_t ts_n)`, which will calculate the total travel time of a group of characters through a path (array) of terrain types. For each terrain type in the array of terrain types, you must calculate the travel speed of the group of characters. You will return the sum total of all travel speeds for all terrain types in the array.

(4 marks)

8. After finishing step 7, the `CHECK_TS3` test cases should pass.
9. Add an implementation of `void remove_slowest(character const* cs, character* out, size_t n, terrain_type const* ts, size_t ts_n)`. `cs` is an array of characters, of length `cs_n`. `ts` is an array of terrain squares of length `ts_n`. `out` is an array of length `cs_n-1`, which your function will populate with characters. The purpose of this function is to determine which character is the slowest, and remove that character. The function will then populate the `out` array with all of the remaining characters (in any order). To determine which character is the slowest, you need to calculate the travel speed with the given array of terrain squares. Note that this function is quite difficult and is only worth 2 marks, so please leave it until last. It is expected that only the top students will be able to complete this function perfectly. (2 marks)

Total: 19 marks.