| **Best Multi-eSports Player**  A videogame company has contracted you to create a program to calculate the Best multi-esport player of each of its tournaments.  Right now, the video games played are League of Legends and Valorant. They plan to add more video games in the future.  You will receive a set of files, each one containing the stats of one game. Each file will start with a row indicating the videogame it refers to.  Each player is assigned a unique nickname  Each file represent a single game  The Best multi-esport player is the player with the most rating points, adding the rating points in all games.  A player will receive 10 additional rating points if their team won the game. Every game must have a winning team. One player may play in different teams and positions in different games, but not in the same game.  The program responsible for generating the files has a bug, that can be reflected in the wrong files format. If one file is wrong, the whole set of files is considered to be wrong and the Best multi-esport player won't be calculated.  **League of Legends**  Each row will represent one player stats, with the format:  player name, nickname, team name; winner; position; kills; deads; assists; damage deal; heal deal  The winning team is the one whose players has the winner column = true (all of them).  KDA is calculated as shown:  (Kills + Assists) / Deaths = KDA  The table below details the rating points each player in a League of Legends game receives depending on his/her position:   |  | KDA | Damage Deal | Heal Deal | | --- | --- | --- | --- | | Top (T) | (Kills + Assists) / Deaths | 0.03 | 0.01 | | Bottom (B) | (Kills + Assists) / Deaths | 0.03 | 0.01 | | Mid (M) | (Kills + Assists) / Deaths | 0.03 | 0.01 | | Jungle (J) | (Kills + Assists) / Deaths | 0.02 | 0.02 | | Support (S) | (Kills + Assists) / Deaths | 0.01 | 0.03 |   *E.g. a player playing as a Mid with 10 kills, 5 deaths and no assists will be granted with 2 KDA points ((10 + 0) / 5 ). Aggregating 2000 damage deal and 200 of healing (2 + 2000\*0.03 + 200\*0.01), the final result is 10 rating points.*  *Example:*  *LEAGUE OF LEGENDS*  *player 1;nick1;Team A;true;T;10;5;2;2000;200*  *player 2;nick2;Team B;false;S;2;10;500;4000*  As file validation, the Team A Kills should match the Team B Deaths, and vice versa.  **Valorant - Team Deathmatch**  Each row will represent one player stats, with the format:  player name, nickname, team name, kills, deaths  The winning team is the one with more kills in total.  KD is calculated as shown:  Kills / Deaths = KDR (0 Deaths is not a valid value).  The table below details the rating points each player in a Valorant game receives:   |  | KDR | | --- | --- | | Player | Kills / Deaths |   *E.g. a player with 10 kills and 2 deaths will be granted with 5 rating points.*  *Example:*  *VALORANT*  *player 1;nick1;Team A;10;2*  *player 2;nick2;Team B;5;7*  *player 3;nick3;TeamB; 7;4*  *player 4;nick4;Team A;1;5*  As file validation, the Team A Kills should match the Team B Deaths, and vice versa.  **What we look at**  You have a fixed time to complete the test (it will depend on the position you are applying for).  You can use any programming language you want (take into account that you are probably applying to a specific language position).  **No UI or database access code is needed.**   * This task is designed to give us an idea of how you think when faced with a very limited amount of time to solve a task of significant complexity. * We are interested in how you structure your code so that it's easily extendable, complies with best OO practices, and easy to modify /understand by others. * We are also interested in seeing how efficient the sorting algorithm you implement is. * We do like tests   **Hand it**  Hand in your solution along with any notes, comments, and assumptions you have made while working on the solution.  Please upload your code to a public or private GIT repository  Enjoy & good look! |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |