**The Project**

This is a data management project with a graphical display. You are to write a program for managing a badminton ladder, which is a form of badminton competition where the players are placed in an ordered list and challenge players higher in the order. Here are the rules in operating the ladder

1. A player on the ladder may challenge another player up to three places above. A challenge to players beyond the three places above is forbidden.

2. If the challenger wins, then s/he moves into the position of the challenged. The challenged and the players in between each moves down the ladder by one place. Other players are not affected

3. If the challenger loses, the ladder remains unchanged.

4. A new player joins the ladder at the bottom.

5. When a player leaves, everyone below moves up by one place to occupy the vacated spot.

initially, the ladder is populated randomly. After numerous challenges, the ladder would assume an order of strength. The picture on the right shows a physical rack for manual recording of a badminton ladder. This project aims to provide an online replacement, which also offers a lot more facilities.

**The Program**

Your program manages the ladder and displays useful information. This means your program must

i. Provide means for a player to take one of these actions’

an Issue a challenge (stating opponent and play-by date)

b. Record the result of a challenge

c. Min the ladder

d. Withdraw from the ladder

e. Make a query

ii. Provide means of recording and displaying the result of a challenge.

iii. Adjust the ladder according to the result.

iv. Register a new player and place him/her on the ladder.

v. Accept the withdrawal of a player from the ladder and update the ladder accordingly.

Your program must produce a graphical display with these elements (imagine this is where players go Mr their information and conduct ladder business):

i. The ladder, with the first player on top and the last player at the bottom.

ii. Facilities for a player to take any action as stated above.

III.A display showing the yet-to-play challenges.

iv. A means for making queries and displaying the answers, your program should inc Queries.

a. The order of the ladder on a specific date

b. The data of a specific challenge based on the names of the players

c. The data of a specific challenge based on the date. The list of matches a player has played, with all the associated data

e. The most active player, i.e. the player with the most matches

f. The least active player, i.e. the player with the least matches

g. The list of matches played in a specific date range

h. You may add more queries as you see fit

Furthermore,

v. The data of the ladder are stored in the two data files specified below, and they must be updated as events take place.

vi. When the program runs, the display should show the current ladder order as well as the results of the latest n challenges (you may choose n, n >r 2).

Within reason, you may choose your own design of this display and what to include in it.

**The Data Files**

The data of the ladder are in two files which are to be updated every time an action is taken. One file, ladder.txt, contains the current order of the ladder, which is an ordered list of the names of the players. The second file, data.txt, contains data on every challenge and the coming and going of players. The data are in chronological order, with the latest data inserted at the end of the file. Specifically, each line of this second file contains one of these three items:

separates the different fields. The names are strings. pl and p2 after the names are integers showing the current positions of the players on the ladder. The date format is OD-MM-YYYY, and the scores format is xx.yy, where ix and yy represent the numerical scores of the challenger and the challenged respectively. A match may have two or three of such scores. Example: CW Lee 5/ff Frost 3/20-07-2000/20-22 18-21

If a score is missing from such a line, it means that the challenge has been issued but the match has not been played, and the date in the line is the play-by date. If this date is in the past. it means that the challenged has not taken up the challenge and therefore forfeits the match; the challenger is automatically the winner.

(Note: Badminton matches are played to the best of three games. The player reaching 21 pants first in a game wins the game. If a game reaches 20 20. then the player leading by two points SubSe’endy wins the game.)

ii. The name of a new player whO pins and the date of joining, with a • sign before the name. Example:-tax Player/18-05-2020

iii. The name of a PlaYer whO leaves and his/her position on the ladder, and the date of leaving, with a- sign before the name. Example: -EX Player :2/18-05-202C

The existing data in ltdder.txt contains the current order of the Ost of players in there. The data in data tat contains the chronological record of the challenges. new players pining and existing players leaving to date. AS your program runs, the data in these files will change.

You may ChOOSe to use yOur own data format. in which Case you most specify this format clearly.

3. **Resources** The main external resource that you will need for this protection is the graphical display. for which you need the graphics Wary provided with PythOn. calked Turtle. Turtle capabilities and regions are described fully in Section 24.1 of the Python documentation, which you can access via the ORA interactive shell by clicking Hs >Python Does and then search for ‘turtle. Torte provides the Funcom required fOr drawings, such as the creators Of ShaPeS and posing of text, and concourse their attributes such as location. color. and more. You should study the different Turtle her: ins that are required kw your graphics There are many Tolle functions, your only need to learn the ones Ourense it is, therefore, necessary that you wort out what you need beforehand

4.**What you need to submit**

1 You need to submit a working Python program that manages the dynamics of a badminton ladder and displays the status appropriately, as stated above

2. The two data Nesgenerated at the end of a sequence of activities, wroth should contain more data than the file first handed out

3. You also need to submit a report in a Word ale, providing

▪ a rude on how to run your program

▪ the data of a sequence of runs of the program. sirloins the (range at each run and the insertion and removal of players. as they come and go

* pictures of your graphics window, with a display of the different information
* highlights the key strengths and limitations of the program