Task 1.4

```
import random
class ChessPlayer:
    def __init__(self, name, age, elo_rating, tenacity, is_boring):
        self.name = name
        self.age = age
        self.elo_rating = elo_rating
        self.tenacity = tenacity
        self.is_boring = is_boring
        self.tournament_score = 0
def simulateMatch(player1, player2):
    elo_diff = abs(player1.elo_rating - player2.elo_rating)
    if elo_diff > 100:
        if player1.elo_rating > player2.elo_rating:
            player1.tournament_score += 1
        else:
            player2.tournament_score += 1
    elif 100 >= elo_diff >= 50:
        random_factor = random.randint(1, 10)
        if(player1.elo_rating < player2.elo_rating):
          lower_elo_product = player1.tenacity * random_factor
          if lower_elo_product > player2.elo_rating:
              player1.tournament_score += 1
          else:
              player2.tournament_score += 1
        else:
          lower_elo_product = player2.tenacity * random_factor
          if lower_elo_product > player1.elo_rating:
              player2.tournament_score += 1
          else:
```

```
            player1.tournament_score += 1
    elif elo_diff < 50:
        if player1.tenacity > player2.tenacity:
            player1.tournament_score += 1
        else:
            player2.tournament_score += 1
    if (player1.is_boring or player2.is_boring) and elo_diff <= 100:
        player1.tournament_score += 0.5
        player2.tournament_score += 0.5
num_players = int(input("Enter the number of players: "))
players = []
for i in range(num_players):
    print(f"\nPlayer {i+1} details:")
    name = input("Name: ")
    age = int(input("Age: "))
    elo_rating = float(input("ELO Rating: "))
    tenacity = int(input("Tenacity: "))
    is_boring = input("Is player boring? (True/False): ").lower() == "true"
    players.append(ChessPlayer(name, age, elo_rating, tenacity, is_boring))

# Simulating matches
for i in range(len(players)):
    for j in range(i + 1, len(players)):
        simulateMatch(players[i], players[j])
        simulateMatch(players[j], players[i])

# Printing tournament results
print("Tournament Results:")
print("{:<10}      {:<10}".format("Name", "Score"))
for player in players:
    print("{:<10}  {:<10}".format(player.name, player.tournament_score))
```