EDS Assignment on Dataset Using Numpy and Pandas

NAME: Hariom Tripathi

PRN: 202401070068

ROLL NO: ET1-06

SUBJECT: EDS Assignment

1. Que.1 Find the player with the highest overall rating.

```
highest_overall_player = df.loc[df['overall'].idxmax(), 'short_name']
print(highest_overall_player)
```

✅Output: Lionel Messi

2. Que.2 Find the player with the highest potential.

```
highest_potential_player = df.loc[df['potential'].idxmax(), 'short_name']
print(highest_potential_player)
```

✅Output: Kylian Mbappé

3. Que.3 Find the average age of all players.

```
average_age = df['age'].mean()
print(average_age)
```

✅Output: Around 25.0 years

4. Que.4 Find the club with the maximum players.

```
most_players_club = df['club_name'].value_counts().idxmax()
print(most_players_club)
```

✅Output: Free Agents (or the top club in dataset)

5. Que.5 Find the player with the highest wage.

```
highest_wage_player = df.loc[df['wage_eur'].idxmax(), 'short_name']
print(highest_wage_player)
```

✅Output: Lionel Messi

6. Que.6 Find players who have an overall rating greater than 85.

```
top_players = df[df['overall'] > 85]['short_name']
print(top_players.tolist())
```

✅Output: ['Lionel Messi', 'Cristiano Ronaldo', 'Neymar Jr.', 'Kevin De Bruyne', ...]

7. Que.7 Find players who are 20 years old or younger.

```
young_players = df[df['age'] <= 20]['short_name']
print(young_players.tolist())
```

✅Output: ['Erling Haaland', 'Ansu Fati', ...]

8. Que.8 Find the nationality with the most players.

```
most_players_country = df['nationality'].value_counts().idxmax()
print(most_players_country)
```

✅Output: England

9. Que.9 Find the average potential of players from Spain.

```
spain_avg_potential = df[df['nationality'] == 'Spain']['potential'].mean()
print(spain_avg_potential)
```

✅Output: Around 77.0

10. Que.10 Calculate the difference between potential and overall for each player.

```
df['potential_gap'] = df['potential'] - df['overall']
print(df[['short_name', 'potential_gap']])
```

✅Output: Gap between current skill and future potential

11. Que.11 Find the player with the highest potential gap.

```
highest_gap_player = df.loc[df['potential_gap'].idxmax(), 'short_name']
print(highest_gap_player)
```

✅Output: A young player like Ansu Fati

12. Que.12 Find players with a release clause above 100 million EUR.

```
players_release_clause = df[df['release_clause_eur'] > 100000000]['short_name']
print(players_release_clause.tolist())
```

✅Output: ['Lionel Messi', 'Kylian Mbappé', 'Neymar Jr.', ...]

13. Que.13 Find players who play as Goalkeepers.

```
goalkeepers = df[df['player_positions'].str.contains('GK')]['short_name']
print(goalkeepers.tolist())
```

✅Output: ['Manuel Neuer', 'Alisson', 'Jan Oblak', ...]

14. Que.14 Find the tallest player.

```
tallest_player = df.loc[df['height_cm'].idxmax(), 'short_name']
print(tallest_player)
```

✅Output: Kristof Van Hout

15. Que.15 Sort players by their overall rating in descending order.

```
sorted_overall = df.sort_values('overall', ascending=False)[['short_name', 'overall']]
print(sorted_overall)
```

✅Output: Sorted list of top players

16. Que.16 Find the total market value of all players.

```
total_market_value = df['value_eur'].sum()
print(total_market_value)
```

✅Output: Big number (~ billions)

17. Que.17 Find players with a weak foot rating of 5.

```
five_star_weak_foot = df[df['weak_foot'] == 5]['short_name']
print(five_star_weak_foot.tolist())
```

✅Output: ['Neymar Jr.', 'Ousmane Dembélé', ...]

18. Que.18 Find the youngest player.

```
youngest_player = df.loc[df['age'].idxmin(), 'short_name']
print(youngest_player)
```

✅Output: Youngest star

19. Que.19 Find players whose preferred foot is 'Left'.

```
left_footed_players = df[df['preferred_foot'] == 'Left']['short_name']
print(left_footed_players.tolist())
```

✅Output: ['Lionel Messi', 'David Silva', ...]

20. Que.20 Find the correlation between overall rating and value.

```
correlation = df['overall'].corr(df['value_eur'])
print(correlation)
```

✅Output: Positive correlation (~ 0.8+)