Exercise 3 (10 points) – can be done individually or in pair

- The first lines of all source files must be comment containing names & IDs of all members. Also create file readme.txt containing names & IDs of all members.
- Put all files (source, input, output) in folder Ex3_xxx where xxx = your full ID. That is, your source files must be in package Ex3_xxx and input/output files (if there is any) must be read from/write to this folder. From now on, you'll get point deduction for wrong package & folder structure.
- The group representative zips Ex3_xxx & submits it to Google Classroom. The other members submit only readme.txt. Email submission is not accepted.
- The exercise is graded only once, and after graded, members can't be added.

==========

1. Copy class Player to your source file. This class must not be changed at all.

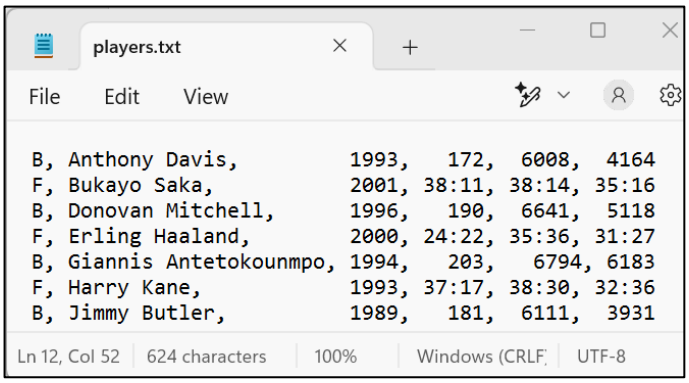
```
class Player {
  public static final int CURRENT_YEAR = 2025;

  private   String name;
  protected int    birthyear, age;

  public Player(String nm, int by)      { name = nm; birthyear = by; }
  public String getName()               { return name; }
  public void printPersonalData()       { /* override this in child class */ }
  public void printStat()               { /* override this in child class */ }
}
```

2. Write classes FootballPlayer and BasketballPlayer that extend Player. Add at least the following variables.
   - FootballPlayer : array of games, array of goals, avgGoals
   - BasketballPlayer : totalGames, totalMins, totalPts, avgMins, avgPts
   - Other variables & methods can be added to these classes

3. Write another class that acts as the main class. In its main method,
   3.1 Create an array of 12 Player objects, e.g. Player [] allPlayers

   3.2 Read each line of input file into allPlayers[i]. The type of allPlayers[i] in each line may be FootballPlayer or BasketballPlayer.

   - Col 0 = type
     F = FootballPlayer
     B = BasketballPlayer
   - Col 1 = player name
   - Col 2 = birth year
   - For FootballPlayer
     Col 3 = games:goals in season 21-22
     Col 4 = games:goals in season 22-23
     Col 5 = games:goals in season 23-24
   - For BasketballPlayer
     Col 3 = total games in seasons 22, 23, 24
     Col 4 = total appearance (mins) in seasons 22, 23, 24
     Col 5 = total points in seasons 22, 23, 24



Note 1 – Read the whole line into a String (e.g. line) and split it at comma. Trim spaces before converting String to number; otherwise, you'll get runtime exception.

```
String line    = scan.nextLine();
String [] cols = line.split(",");
int year       = Integer.parseInt( cols[2].trim() );
```

– You can also split games:goals into 2 values at colon.

```
String [] s    = col[3].split(":");     // s[0] keeps games, s[1] keeps goals
```

Note 2 – To compare Strings

```
if (type.equals("F"))                   // create FootballPlayer object
if (type.equalsIgnoreCase("F"))         // create FootballPlayer object
```

or use switch-case

4. After reading all data into array allPlayers

4.1 For both type of objects -> calculate player age (until 2025) from birth year. Print all player names, birth years, ages in the reverse order of the input.

4.2 For FootballPlayer -> calculate total games, total goals, average goals per game. Print player names, total games, total goals, average goals, goals in last season (i.e. 23-24). The output order must follow the original input order.

4.3 For BasketballPlayer -> calculate average appearance and average points per game. Print player names, total games, total mins & average mins, total points & average points. The output order must follow the original input order.

4.4 In 4.1-4.3, all numbers must be printed right aligned. All average values must be printed in 2 decimal places.

Note 3 – To check type of object

```
if ( allPlayers[i] instanceof FootballPlayer ) {
   FootballPlayer p = (FootballPlayer) allPlayers[i];
   p.method();
}
```

```
--- exec:3.1.0:exec (default-cli) @ solutions ---
=== All player data (by reverse order) ===
Son Heung-Min             born 1992    age = 33
Nikola Jokic              born 1995    age = 30
Mohamed Salah             born 1992    age = 33
Marcus Rashford           born 1997    age = 28
LeBron James              born 1984    age = 41
Jimmy Butler              born 1989    age = 36
Harry Kane                born 1993    age = 32
Giannis Antetokounmpo     born 1994    age = 31
Erling Haaland            born 2000    age = 25
Donovan Mitchell          born 1996    age = 29
Bukayo Saka               born 2001    age = 24
Anthony Davis             born 1993    age = 32

=== Football player statistics (by input order) ===
Bukayo Saka               total games = 111    total goals =  41 (0.37 per game)     last season goals = 16
Erling Haaland            total games =  90    total goals =  85 (0.94 per game)     last season goals = 27
Harry Kane                total games = 107    total goals =  83 (0.78 per game)     last season goals = 36
Marcus Rashford           total games =  93    total goals =  28 (0.30 per game)     last season goals =  7
Mohamed Salah             total games = 105    total goals =  60 (0.57 per game)     last season goals = 18
Son Heung-Min             total games = 106    total goals =  50 (0.47 per game)     last season goals = 17

=== Basketball player statistics (by input order) ===
Anthony Davis             total games = 172    total mins = 6008 (34.93 per game)    total points = 4164 (24.21 per game)
Donovan Mitchell          total games = 190    total mins = 6641 (34.95 per game)    total points = 5118 (26.94 per game)
Giannis Antetokounmpo     total games = 203    total mins = 6794 (33.47 per game)    total points = 6183 (30.46 per game)
Jimmy Butler              total games = 181    total mins = 6111 (33.76 per game)    total points = 3931 (21.72 per game)
LeBron James              total games = 182    total mins = 6542 (35.95 per game)    total points = 5107 (28.06 per game)
Nikola Jokic              total games = 222    total mins = 7535 (33.94 per game)    total points = 5779 (26.03 per game)
------------------------------------------------------------------------
BUILD SUCCESS
```