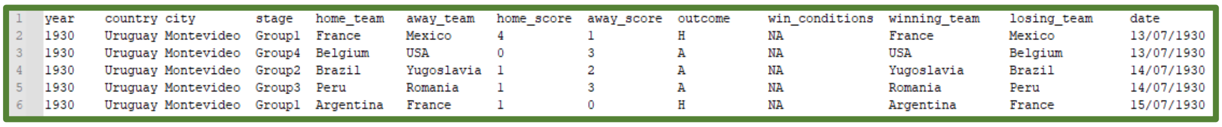
The FIFA World Cup is a global football competition contested by the various football-playing nations of the world. It is contested every four years and is the most prestigious and important trophy in the sport of football. The first World Cup was held in 1930 in Uruguay with the most recent being held in Russia in 2018. The count down to the start of the FIFA World Cup in Qatar on November 20 is well underway.

The starter code contains a file called *matches.txt* which contains information relating to every single World Cup match played from Uruguay in 1930 to Russia in 2018.

A screen-grab of the first five records in the file (along with column headings are shown below).



year	country	city	stage	home_team	away_team	home_score	away_score	outcome	win_conditions	winning_team	losing_team	date
1930	Uruguay	Montevideo	Group1	France	Mexico	4	1	H	NA	France	Mexico	13/07/1930
1930	Uruguay	Montevideo	Group4	Belgium	USA	0	3	A	NA	USA	Belgium	13/07/1930
1930	Uruguay	Montevideo	Group2	Brazil	Yugoslavia	1	2	A	NA	Yugoslavia	Brazil	14/07/1930
1930	Uruguay	Montevideo	Group3	Peru	Romania	1	3	A	NA	Romania	Peru	14/07/1930
1930	Uruguay	Montevideo	Group1	Argentina	France	1	0	H	NA	Argentina	France	15/07/1930

The fields in the file are as follows:

year: The year the world cup was held.

county: The host nation of the competition.

city: The city the match is being played in.

stage: The stage of the tournament (group stage/quarter final/semi-final etc..)

home_team: The designated home team for the game.

away_team: The designated away team for the game.

home_score: The number of goals scored in the game by the designated home team.

away_score: The number of goals scored in the game by the designated away team.

win_conditions: If the winning team needed AET (added extra time) or penalties to win. Note, a value of NA (not applicable) is present for games that did not require AET or penalties.

winning_team: The team that has won the game.

losing_team: The team that has lost the game.

The game may of course have finished as a draw (tie), in which case, there was no winning or losing team and the value of NA will appear in both of the above columns.

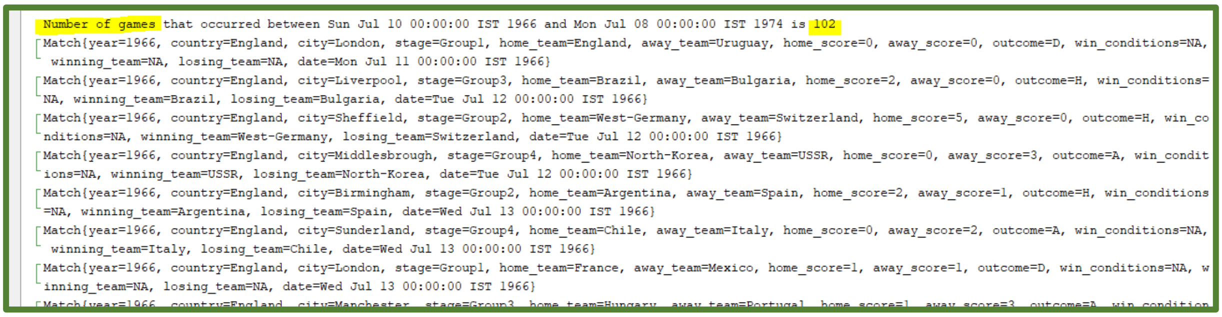
date: The date the match took place (the date is presented in the format dd/mm/yyyyy - for example 1/9/2022).

In the starter code that is available for this assignment, a *Match* class is provided and models a record from the file.

A class called *FileIO* is also provided and contains one public method – *readFile*. This method accepts the file name as an argument and will then read each record from the file, convert it to a *Match* object and then add the object to a list. It is this list of *Match* objects that the method returns.

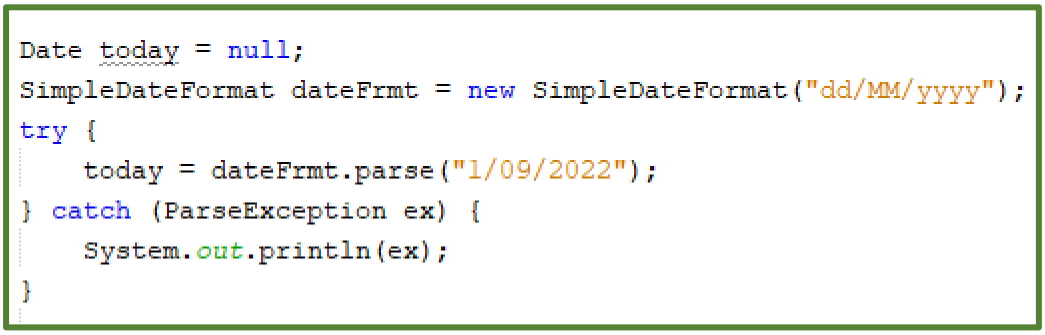
Task ONE

Write and execute a Callable that will accept three arguments - a list of matches, a start date and an end date. The Callable will then determine and return a **list** of match objects which took place between the start and end date. For example, if the start date is 10/07/1966 and the end date is 8/07/1974 a list with 102 objects would be returned. You must then display the contents of the list on the console and an indication of how many games were played in that period (in this case 102).



```
Number of games that occurred between Sun Jul 10 00:00:00 IST 1966 and Mon Jul 08 00:00:00 IST 1974 is 102
Match{year=1966, country=England, city=London, stage=Group1, home_team=England, away_team=Uruguay, home_score=0, away_score=0, outcome=D, win_conditions=NA,
 winning_team=NA, losing_team=NA, date=Mon Jul 11 00:00:00 IST 1966}
Match{year=1966, country=England, city=Liverpool, stage=Group3, home_team=Brazil, away_team=Bulgaria, home_score=2, away_score=0, outcome=H, win_conditions=
NA, winning_team=Brazil, losing_team=Bulgaria, date=Tue Jul 12 00:00:00 IST 1966}
Match{year=1966, country=England, city=Sheffield, stage=Group2, home_team=West-Germany, away_team=Switzerland, home_score=5, away_score=0, outcome=H, win_co
nditions=NA, winning_team=West-Germany, losing_team=Switzerland, date=Tue Jul 12 00:00:00 IST 1966}
Match{year=1966, country=England, city=Middlesbrough, stage=Group4, home_team=North-Korea, away_team=USSR, home_score=0, away_score=3, outcome=A, win_condit
ions=NA, winning_team=USSR, losing_team=North-Korea, date=Tue Jul 12 00:00:00 IST 1966}
Match{year=1966, country=England, city=Birmingham, stage=Group2, home_team=Argentina, away_team=Spain, home_score=2, away_score=1, outcome=H, win_conditions
=NA, winning_team=Argentina, losing_team=Spain, date=Wed Jul 13 00:00:00 IST 1966}
Match{year=1966, country=England, city=Sunderland, stage=Group4, home_team=Chile, away_team=Italy, home_score=0, away_score=2, outcome=A, win_conditions=NA,
 winning_team=Italy, losing_team=Chile, date=Wed Jul 13 00:00:00 IST 1966}
Match{year=1966, country=England, city=London, stage=Group1, home_team=France, away_team=Mexico, home_score=1, away_score=1, outcome=D, win_conditions=NA, w
inning_team=NA, losing_team=NA, date=Wed Jul 13 00:00:00 IST 1966}
Match{year=1966, country=England, city=Manchester, stage=Group3, home_team=Hungary, away_team=Portugal, home_score=1, away_score=3, outcome=A, win_condition
```

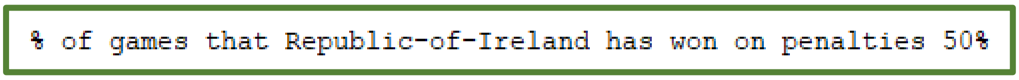
To create a Date, the first thing you need to do is to create a SimpleDateFormat. You can then use this format to initialise a date object: For example:



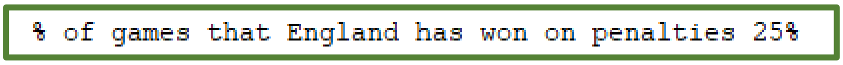
```
Date today = null;
SimpleDateFormat dateFrmt = new SimpleDateFormat("dd/MM/yyyy");
try {
    today = dateFrmt.parse("1/09/2022");
} catch (ParseException ex) {
    System.out.println(ex);
}
```

Task TWO

In the starter code, you will notice an array that contains the names of each nation that has participated in the World Cup since its inception. There are 81 elements/nations in this array. Schedule a Runnable to run every 3 seconds with an initial delay of 2 seconds. The Runnable should generate a random number between 0 and 80 and use this number to subscript the array and select a nation from it. You must then print the percentage of games that the selected nation has won on penalties in the World Cup. For example:



```
% of games that Republic-of-Ireland has won on penalties 50%
```



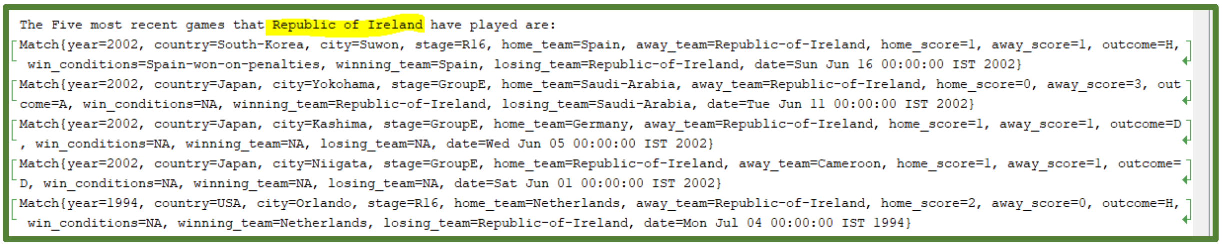
```
% of games that England has won on penalties 25%
```

You are permitted to move this array to a different location in your solution if you wish.

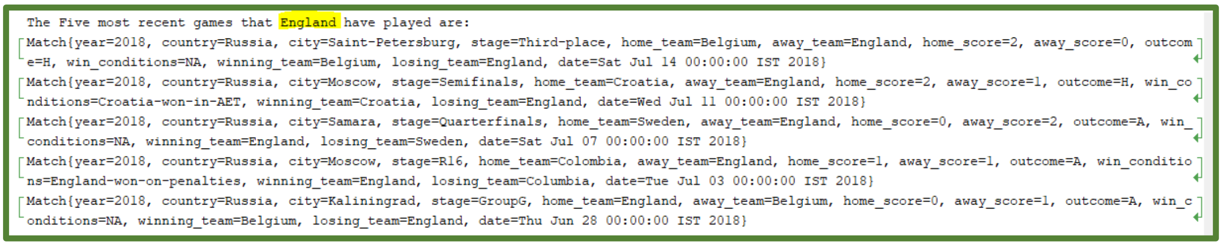
Task THREE

Write and execute a Callable that will return a list of the five most recent games that a given nation has played in the World Cup. The list should be ordered so that the most recent games appear first. You can decide on the type and number of arguments that this Callable accepts yourself.

For example:



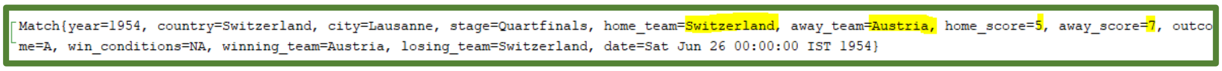
```
The Five most recent games that Republic of Ireland have played are:
Match{year=2002, country=South-Korea, city=Suwon, stage=R16, home_team=Spain, away_team=Republic-of-Ireland, home_score=1, away_score=1, outcome=H,
 win_conditions=Spain-won-on-penalties, winning_team=Spain, losing_team=Republic-of-Ireland, date=Sun Jun 16 00:00:00 IST 2002}
Match{year=2002, country=Japan, city=Yokohama, stage=GroupE, home_team=Saudi-Arabia, away_team=Republic-of-Ireland, home_score=0, away_score=3, out
come=A, win_conditions=NA, winning_team=Republic-of-Ireland, losing_team=Saudi-Arabia, date=Tue Jun 11 00:00:00 IST 2002}
Match{year=2002, country=Japan, city=Kashima, stage=GroupE, home_team=Germany, away_team=Republic-of-Ireland, home_score=1, away_score=1, outcome=D
, win_conditions=NA, winning_team=NA, losing_team=NA, date=Wed Jun 05 00:00:00 IST 2002}
Match{year=2002, country=Japan, city=Niigata, stage=GroupE, home_team=Republic-of-Ireland, away_team=Cameroon, home_score=1, away_score=1, outcome=
D, win_conditions=NA, winning_team=NA, losing_team=NA, date=Sat Jun 01 00:00:00 IST 2002}
Match{year=1994, country=USA, city=Orlando, stage=R16, home_team=Netherlands, away_team=Republic-of-Ireland, home_score=2, away_score=0, outcome=H,
 win_conditions=NA, winning_team=Netherlands, losing_team=Republic-of-Ireland, date=Mon Jul 04 00:00:00 IST 1994}
```



```
The Five most recent games that England have played are:
Match{year=2018, country=Russia, city=Saint-Petersburg, stage=Third-place, home_team=Belgium, away_team=England, home_score=2, away_score=0, outcom
e=H, win_conditions=NA, winning_team=Belgium, losing_team=England, date=Sat Jul 14 00:00:00 IST 2018}
Match{year=2018, country=Russia, city=Moscow, stage=Semifinals, home_team=Croatia, away_team=England, home_score=2, away_score=1, outcome=H, win_co
nditions=Croatia-won-in-AET, winning_team=Croatia, losing_team=England, date=Wed Jul 11 00:00:00 IST 2018}
Match{year=2018, country=Russia, city=Samara, stage=Quarterfinals, home_team=Sweden, away_team=England, home_score=0, away_score=2, outcome=A, win_
conditions=NA, winning_team=England, losing_team=Sweden, date=Sat Jul 07 00:00:00 IST 2018}
Match{year=2018, country=Russia, city=Moscow, stage=R16, home_team=Colombia, away_team=England, home_score=1, away_score=1, outcome=A, win_conditio
ns=England-won-on-penalties, winning_team=England, losing_team=Columbia, date=Tue Jul 03 00:00:00 IST 2018}
Match{year=2018, country=Russia, city=Kaliningrad, stage=GroupG, home_team=England, away_team=Belgium, home_score=0, away_score=1, outcome=A, win_c
onditions=NA, winning_team=Belgium, losing_team=England, date=Thu Jun 28 00:00:00 IST 2018}
```

Task FOUR

Write and execute a Runnable that will accept a single argument – a list of matches. You should then endeavour to identify from this list the match with the highest number of goals scored in Word Cup history (it is a 7-5 win for Austria over Switzerland). You must then print the details of this game to the console.



```
Match{year=1954, country=Switzerland, city=Lausanne, stage=Quartfinals, home_team=Switzerland, away_team=Austria, home_score=5, away_score=7, outco
me=A, win_conditions=NA, winning_team=Austria, losing_team=Switzerland, date=Sat Jun 26 00:00:00 IST 1954}
```

Marks	
Task 1	20 Marks
Task 2	30 Marks
Task 3	10 Marks
Task 4	30 Marks

All four tasks are to be submitted to a suitable Executor service and are to run concurrently.