Main concepts:

- An **action** is a set of sentences grouped together and oriented to do something.

-An **action** is enclosed between brackets {   } and has a similar signature as a function except that it does not return anything.

-An **action** can also receive parameters and have local variables.

-To declare an action you have to set ***void*** as the return type and no return value appears within the action.

Example 1: Actions that are already known from the very beginning of FU1:

a)Console.WriteLine (.....)

Console.Write (.....)

Console.Clear( )

b)
```
public static void Main( string[] args)
   {
       // this action does what we need
   }
```

Example 2: An action that shows options of a menu:

```
public static void ShowOptions( )
{
  Console.Clear( );
  Console.WriteLine("1 - CUSTOMERS");
  Console.WriteLine("2 - PROVIDERS");
  Console.WriteLine("3 - INVOICES");
  Console.WriteLine("0 - EXIT");
}
```

An action is called from another action or function, except the Main action, which is called from the IDE or the command line (operating system).

```
public static void Main( string[] args)

{   int op;

    ShowOptions( );

    op = ReadIntegerFromInterval(0,3);

    Console.WriteLine(“YOU HAVE SELECTED +op + “OPTION);

}
```

PUTTING EVERYTHING TOGETHER: Follow the project GameCenter

We are going to build a menu with different options by using **actions**, **functions**, **exceptions**, our brand new loop **do/while** and the **ConsoleKeyInfo** struct:

Main action: Calls to different Actions:

MostrarMenu : just to display the different options available.

MsgNextScreen and uses *ConsoleKeyInfo* data type and *ConsoleKey* enum to detect keystrokes.

DoJugarCaraOCreu: Action that lets user play one game.

DoJugarDaus: Action that lets user play another game.

And both actions call a function to enter an integer value between a min and a max value. This function is called ***public static int LlegirEnterEntre(int min, int max)***

Look at the solution of this exercise for more detail.

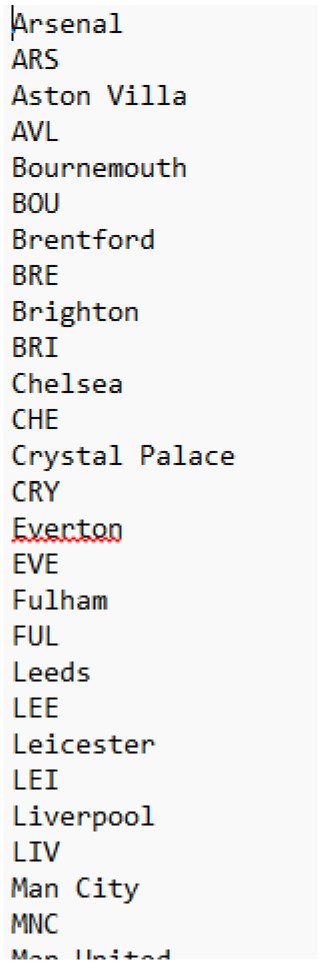
What to do?

You have to create a new solution that uses 2 files that contain teams (TEAMS.TXT) and matches played during the 2022-2023 season of the Premier League (MATCHES.TXT).

FORMAT OF TEAMS.TXT: Each element contains 2 lines :

First line is the team name

the second line is the team abbreviation.



```
Arsenal
ARS
Aston Villa
AVL
Bournemouth
BOU
Brentford
BRE
Brighton
BRI
Chelsea
CHE
Crystal Palace
CRY
Everton
EVE
Fulham
FUL
Leeds
LEE
Leicester
LEI
Liverpool
LIV
Man City
MNC
Man United
```

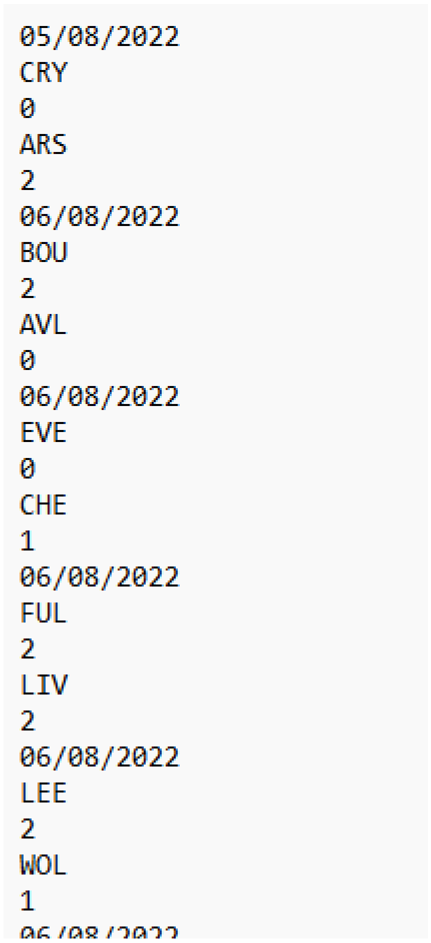
FORMAT OF MATCHES.TXT

One element contains 5 different lines:

- **Date of match**
- **Abbreviation of the Home team**
- **Goals scored by the home team**
- **Abbreviation of the away team**
- **Goals scored by the away team**

Here you have a short sample of the file:(FIRST ELEMENT IS

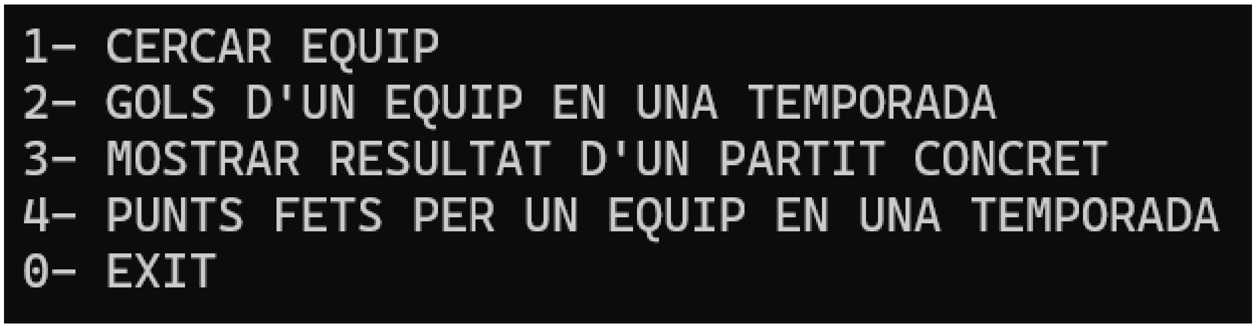
CRYSTAL PALACE 0 ARSENAL 2 ON the 5th of August, 2022



```
05/08/2022
CRY
0
ARS
2
06/08/2022
BOU
2
AVL
0
06/08/2022
EVE
0
CHE
1
06/08/2022
FUL
2
LIV
2
06/08/2022
LEE
2
WOL
1
06/08/2022
```

You have to implement a menu in the main program with 5 options:

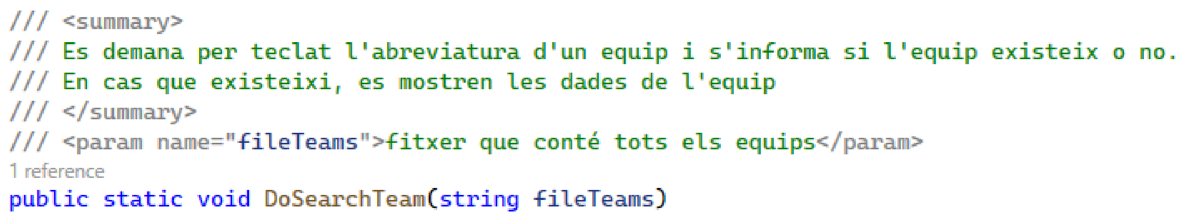
(you also need MessageNextScreen action)



```
1- CERCAR EQUIP
2- GOLS D'UN EQUIP EN UNA TEMPORADA
3- MOSTRAR RESULTAT D'UN PARTIT CONCRET
4- PUNTS FETS PER UN EQUIP EN UNA TEMPORADA
0- EXIT
```

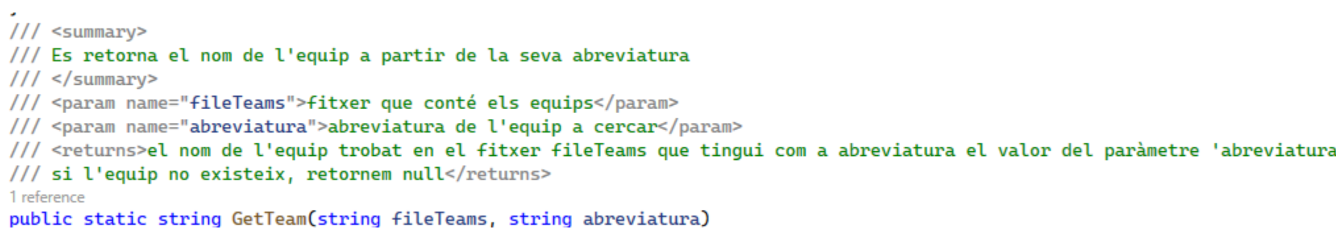
OPTION 1: DoSearchTeam

The 1st option uses only TEAMS.TXT and calls the action DoSearchTeam to obtain the name of the team from an abbreviation given by the user.



```
/// <summary>
/// Es demana per teclat l'abreviatura d'un equip i s'informa si l'equip existeix o no.
/// En cas que existeixi, es mostren les dades de l'equip
/// </summary>
/// <param name="fileTeams">fitxer que conté tots els equips</param>
1 reference
public static void DoSearchTeam(string fileTeams)
```

And DoSearchTeam needs to call the following function:

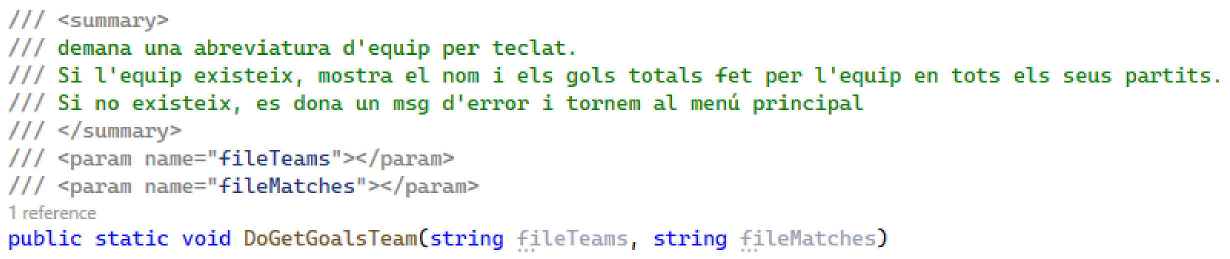


```
/// <summary>
/// Es retorna el nom de l'equip a partir de la seva abreviatura
/// </summary>
/// <param name="fileTeams">fitxer que conté els equips</param>
/// <param name="abreviatura">abreviatura de l'equip a cercar</param>
/// <returns>el nom de l'equip trobat en el fitxer fileTeams que tingui com a abreviatura el valor del paràmetre 'abreviatura
/// si l'equip no existeix, retornem null</returns>
1 reference
public static string GetTeam(string fileTeams, string abreviatura)
```

We will solve this problem all together in class.

OPTION 2: GetGoalsTeam

The 2nd option uses both files (TEAMS.TXT AND MATCHES.TXT) to ask user for a team abbreviation and calculates the total number of goals scored by the team during all the season



```
/// <summary>
/// demana una abreviatura d'equip per teclat.
/// Si l'equip existeix, mostra el nom i els gols totals fet per l'equip en tots els seus partits.
/// Si no existeix, es dona un msg d'error i tornem al menú principal
/// </summary>
/// <param name="fileTeams"></param>
/// <param name="fileMatches"></param>
1 reference
public static void DoGetGoalsTeam(string fileTeams, string fileMatches)
```

OPTION 3 and 4: In the 2nd version of this document.