**Phase-End Project**

**Player and Team Project**

**Application Environment:** Console Application using C#

**Aim:**

FastPace Cricket Academy has decided to create a solution to maintain information about the teams’ players for one day game with the below functionalities:

* User will be able to add a player to the team with details Player Id, Name, and Age.
* User will be able to remove a player from the team by passing the player’s Id.
* User will be able to get player details by passing the player’s Id.
* User will be able to get player details by passing the player’s name.
* User will be able to get all player details.
* User will not be able to add more than 11 players to the team.

**GitHub link:**

https://github.com/Sukirthalakshmanan1/Phase-end-Project

**Procedure:**

Step 1: Create a Visual studio Windows console project. Create Player class,OneDayTeam class and ITeam Interface

Step 2: Create properties for player class.The properties are Player\_Id,Player\_Name,PlayerAge as public.

Step 3: In ITeam interface declare functions .

* void Add(Player player):

To implement functionality for adding a player to the Team by passing an object of type Player as a parameter.

* void Remove(int playerId):

Implement functionality for removing the player from the Team by passing Player Id as a parameter.

* Player GetPlayerById(int playerId):

To implement functionality to get player by passing Player Id as a parameter.

* Player GetPlayerByName(string playerName):

To implement functionality to get player by Player Name as a parameter

* List<Player> GetAllPlayers():

To implement functionality to get all players from the Team

Step 4: Create a derived class **OneDayTeam** to implement ITeam interface functions.

Step 5: Create a constructor OneDayTeam and set capacity to 11.

Step 6: Create a delegate function for add method in OneDayTeam.

* void Add(Player player)

Implement add function get player id,player name,age from the user and add the player and display the message”Added successfully”.

If the player count is 11 then we cannot add the player so display the message ”Already 11 members are there !”

* void Remove(int playerId)

To remove the player get player id from user whom needs to be removed.Here use Find function to find the player id and if the player id is found remove the player id and display the message “Removed successfully” and if player id is not found display the message “Not found !”.

* Player GetPlayerById(int playerId)

To get player by id -get player id from user and using where function (lamda expression) find the player by id given and display the player details.If the player id is not present then display the message “Id not found”.

* Player GetPlayerByName(string playerName)

To get player by name -get player name from user and using where function (lamda expression) find the player by name given and display the player details. If the player name is not present then display the message “Player name not found”.

* List<Player> GetAllPlayers()

Display all the team members(players) in the list.

Step 7:Create a menu to prompt the user to perform the required functions using switch statements.and create object for player class and OneDayTeam class.

Create an extension method in extension2 class and call the method using player class as (object of player class).M4().

Step 8: Call the methods with objects in main .

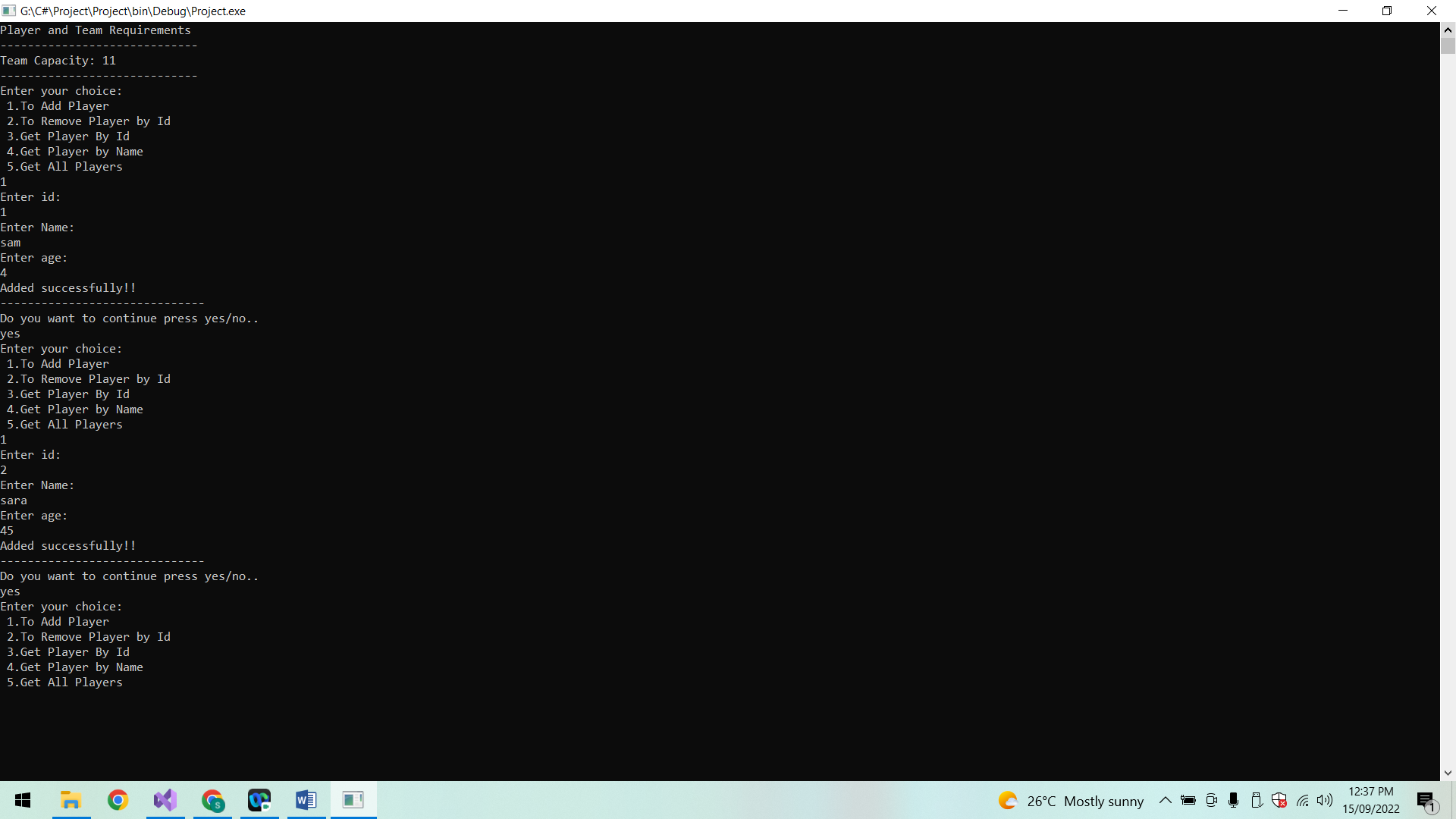
Step 9: create a repository in Github via visual studio

Step 10: Commit all the changes and Push the code to Github.

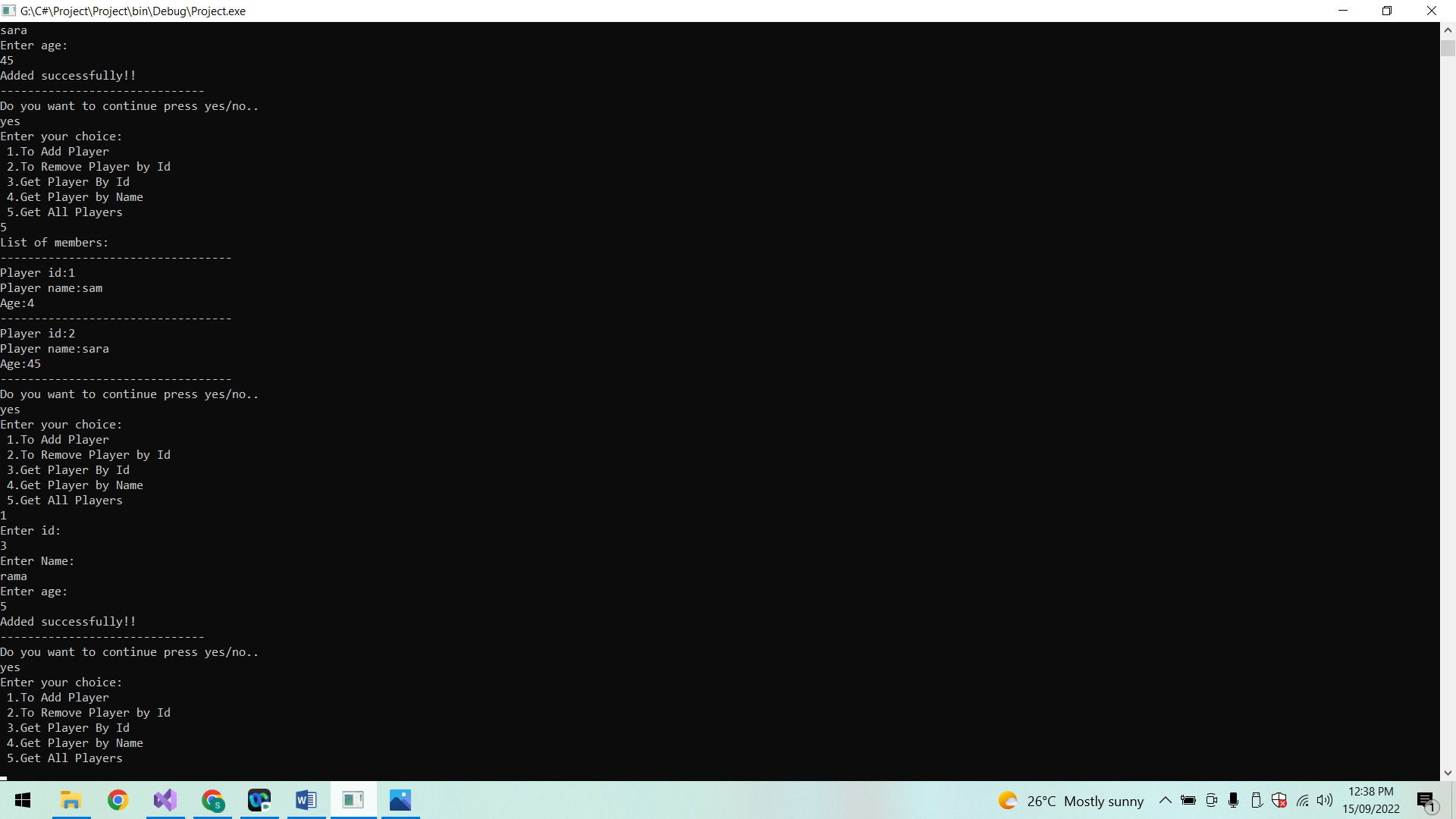
**Output and Screenshots :**

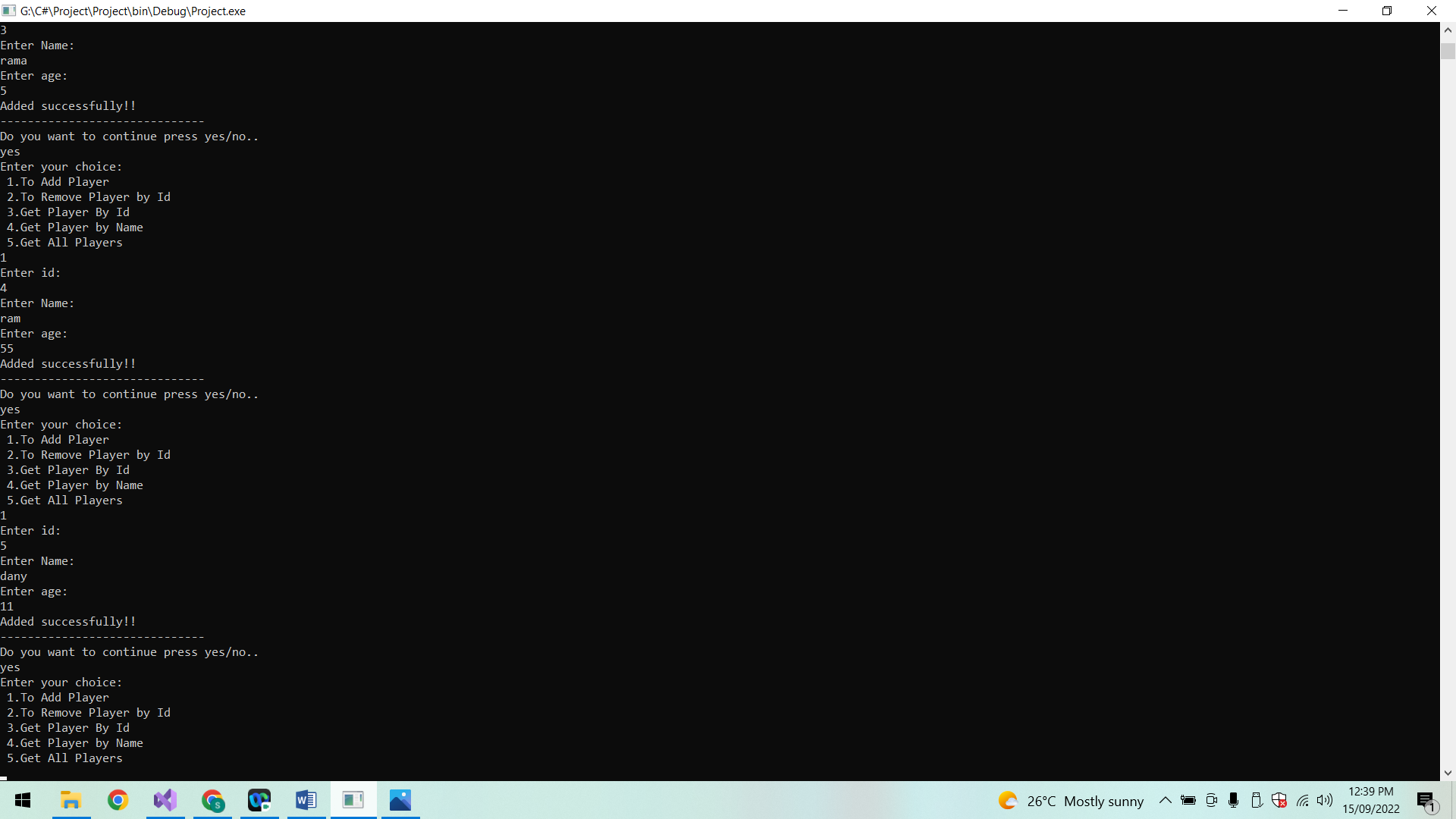
Extension method is implemented and capacity is set to 11

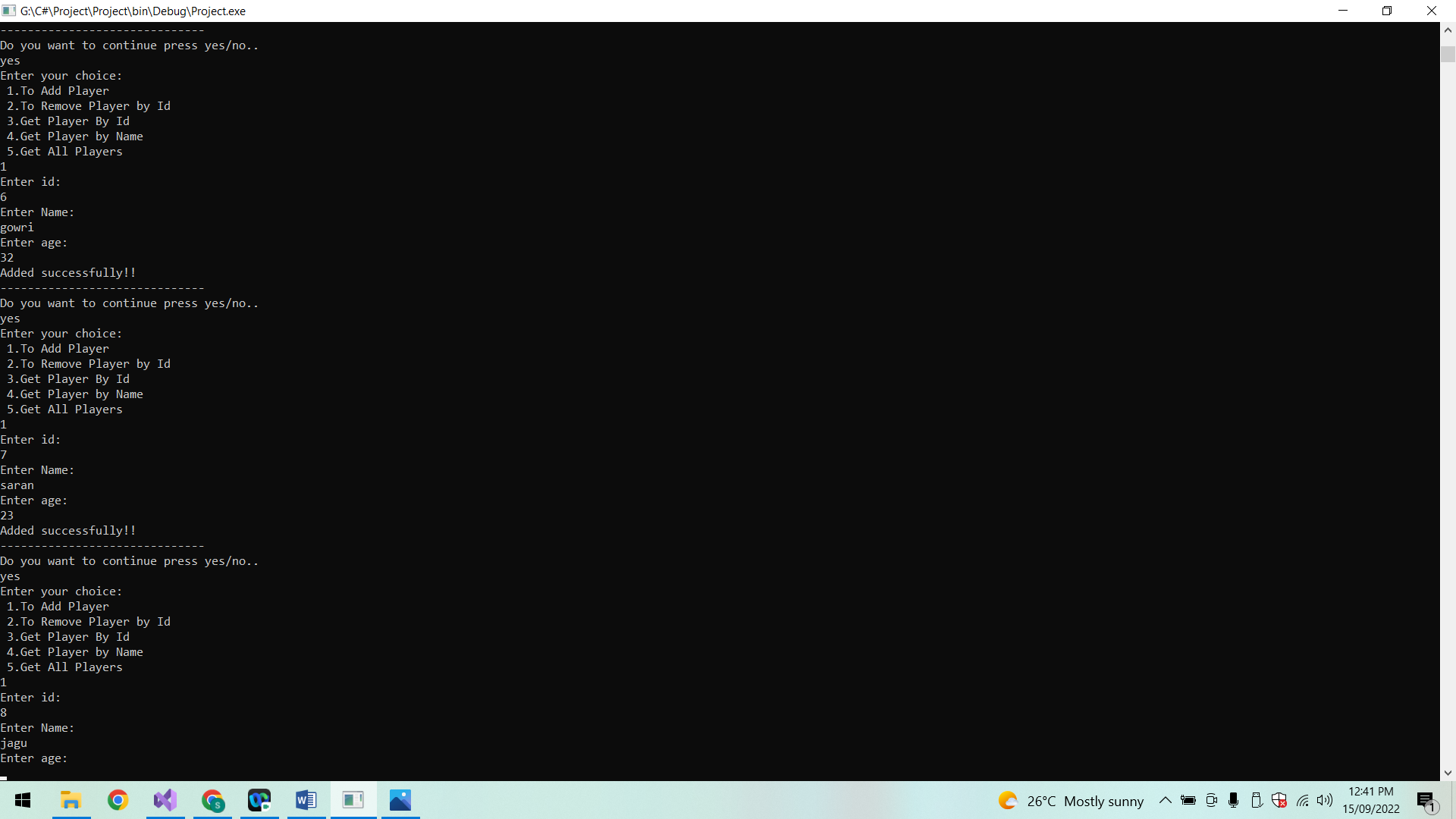
**Add:**

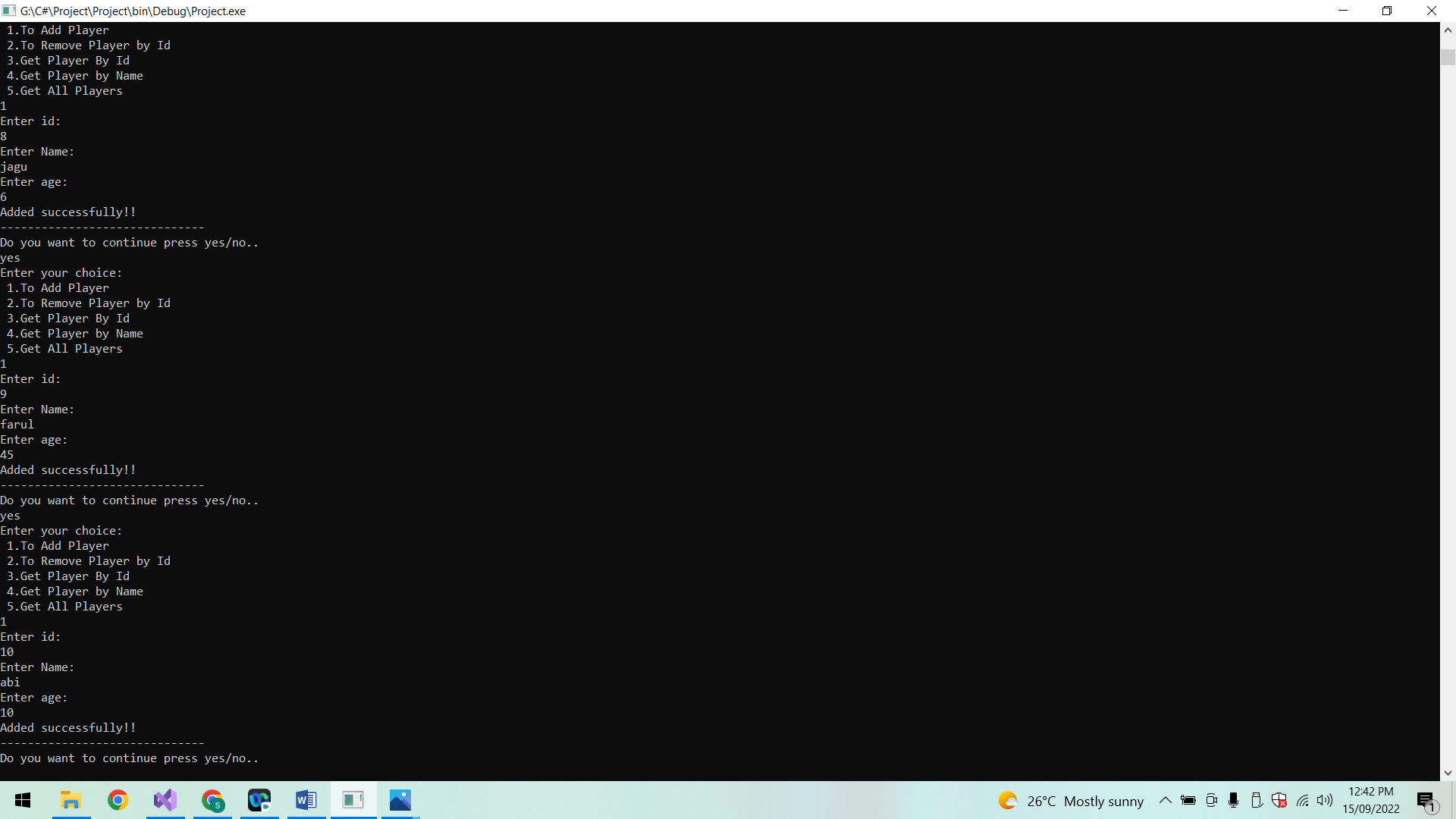
****

**List of Players:**

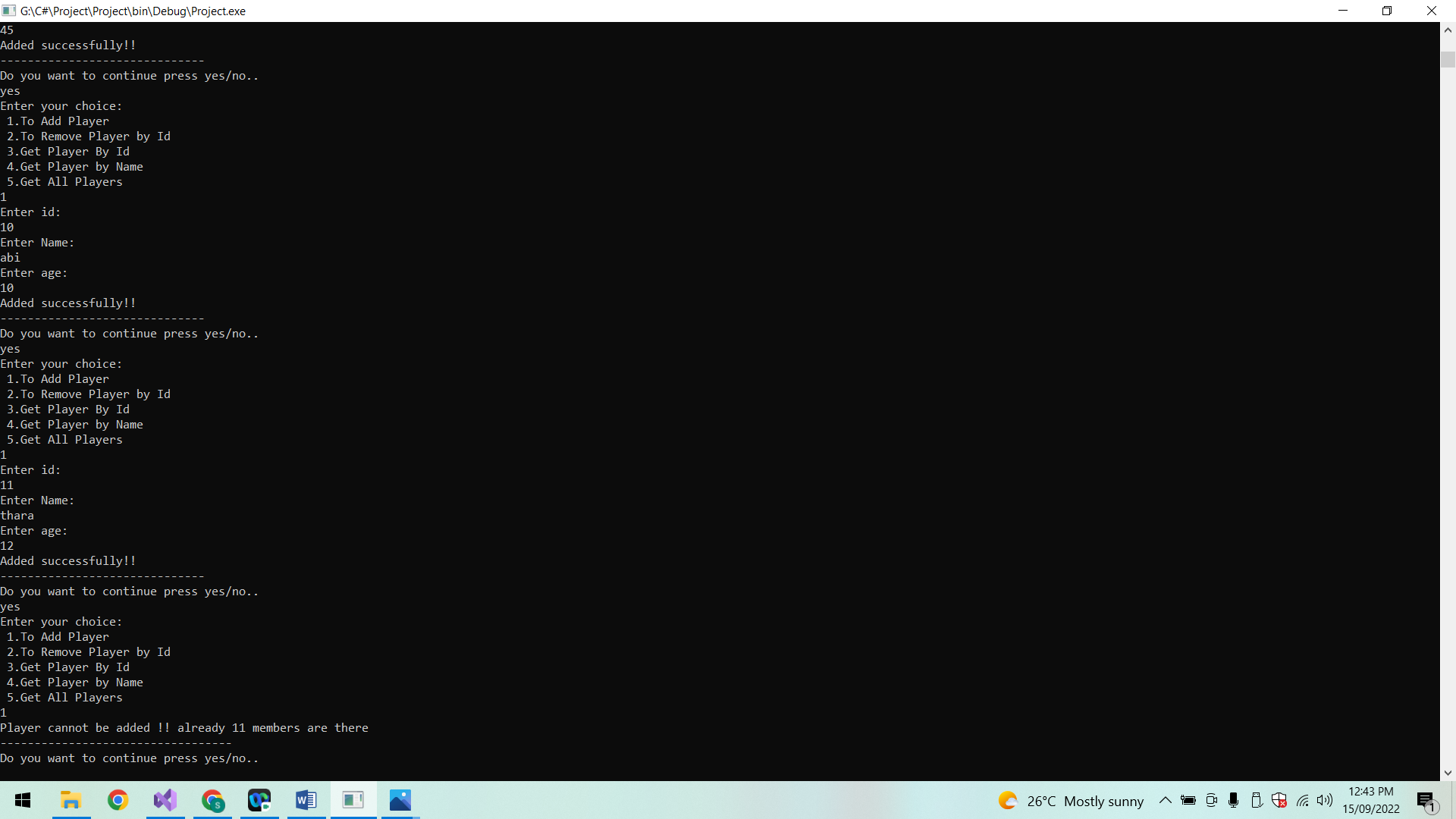
****

****

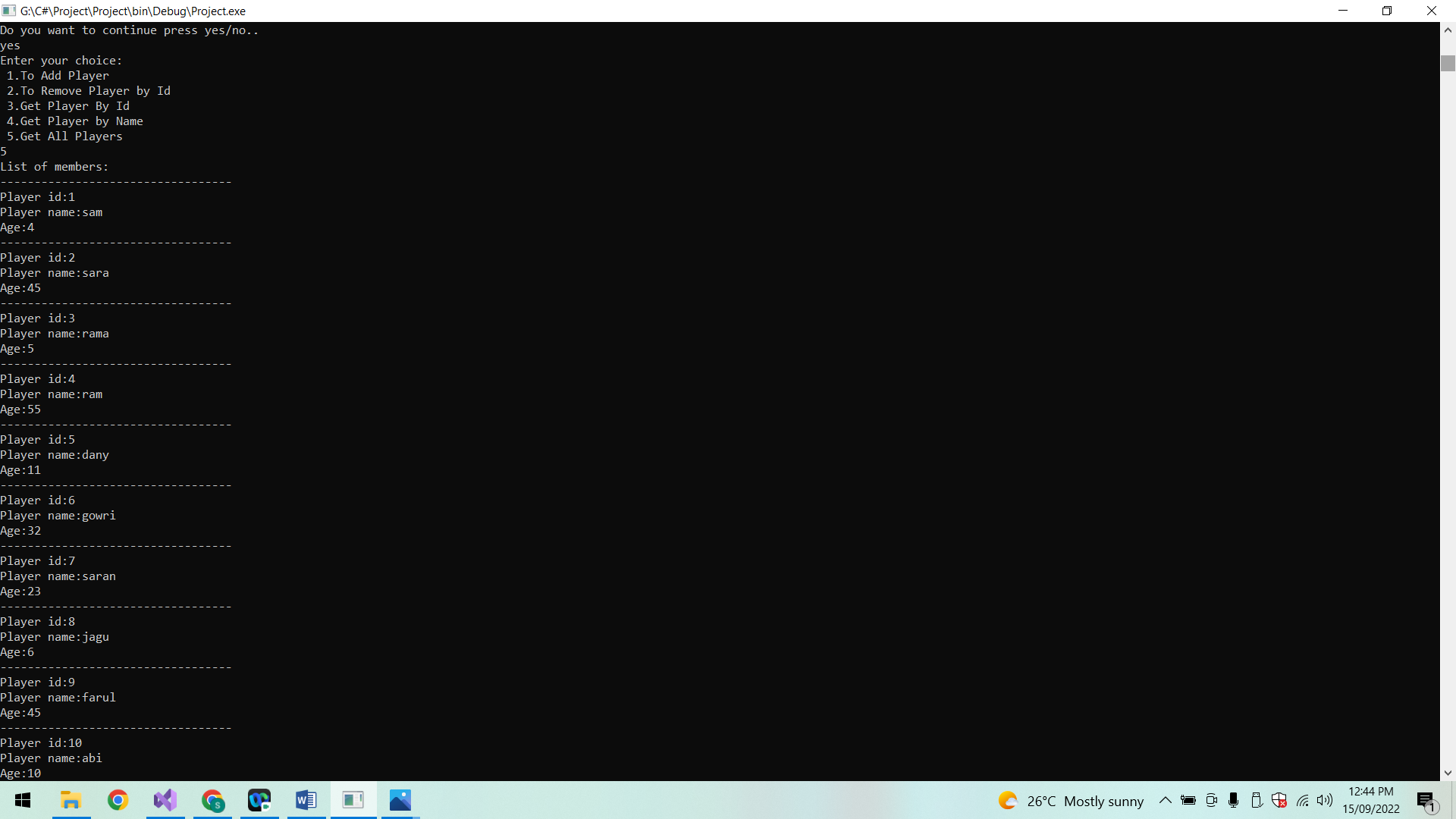
****

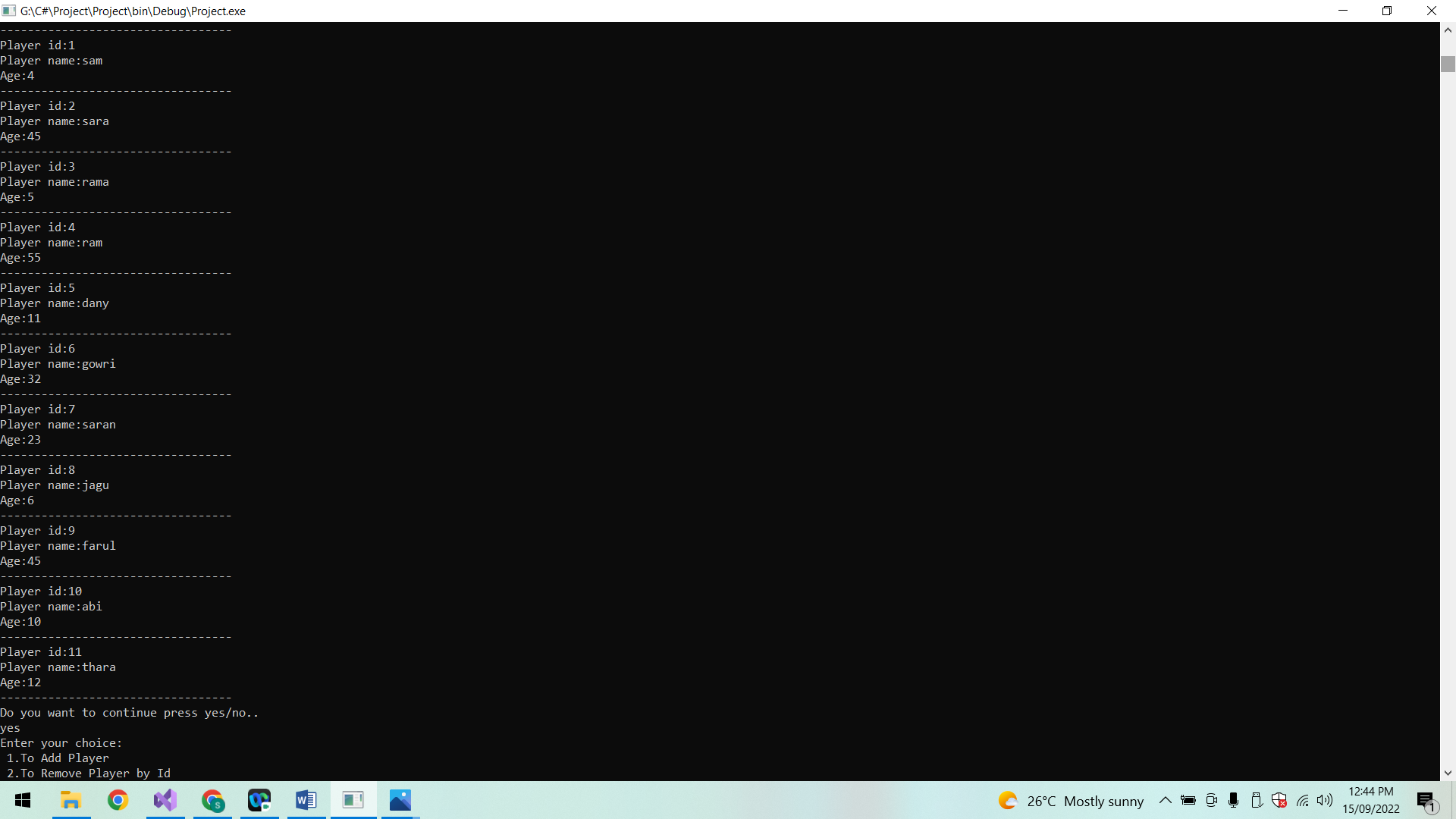
****

After 11 members are added it displays “already 11 members are there player cannot be added”

****

**List of players:**

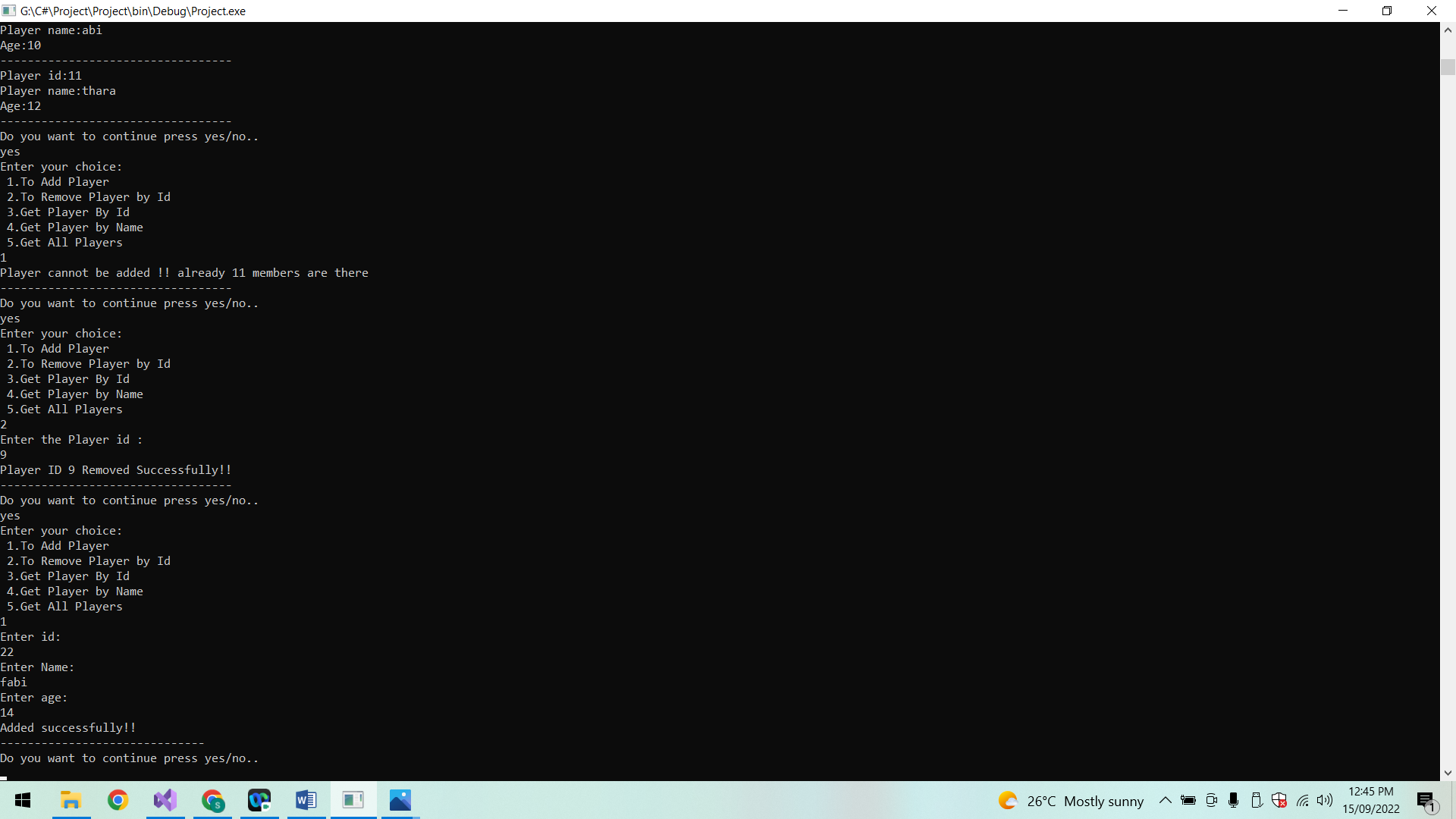
****

****

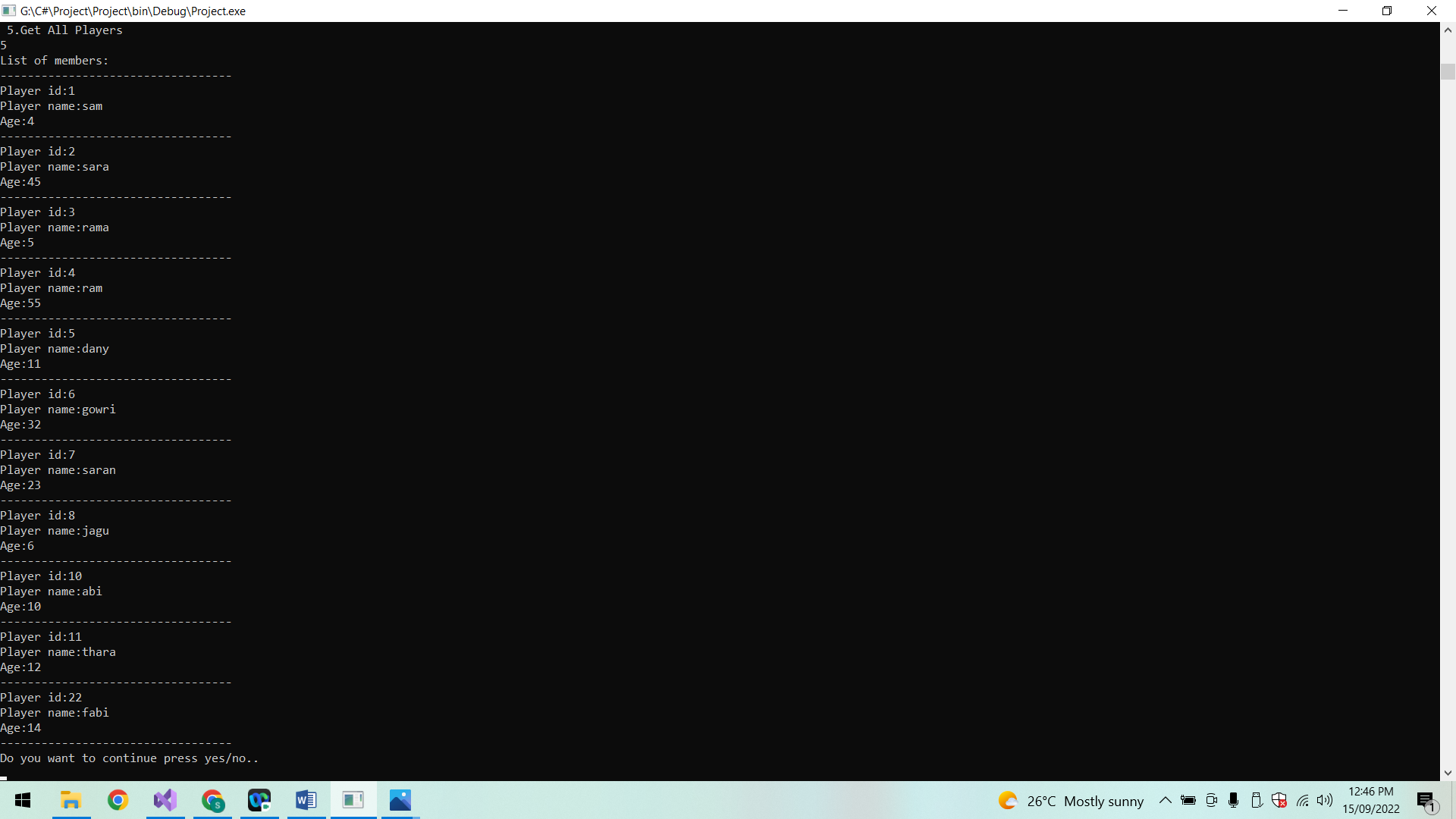
**Remove player using player id:**

If the player id is found remove the player id and displays the message “Removed successfully” and if player id is not found displays the message “Not found !”.

Now other player can be added.

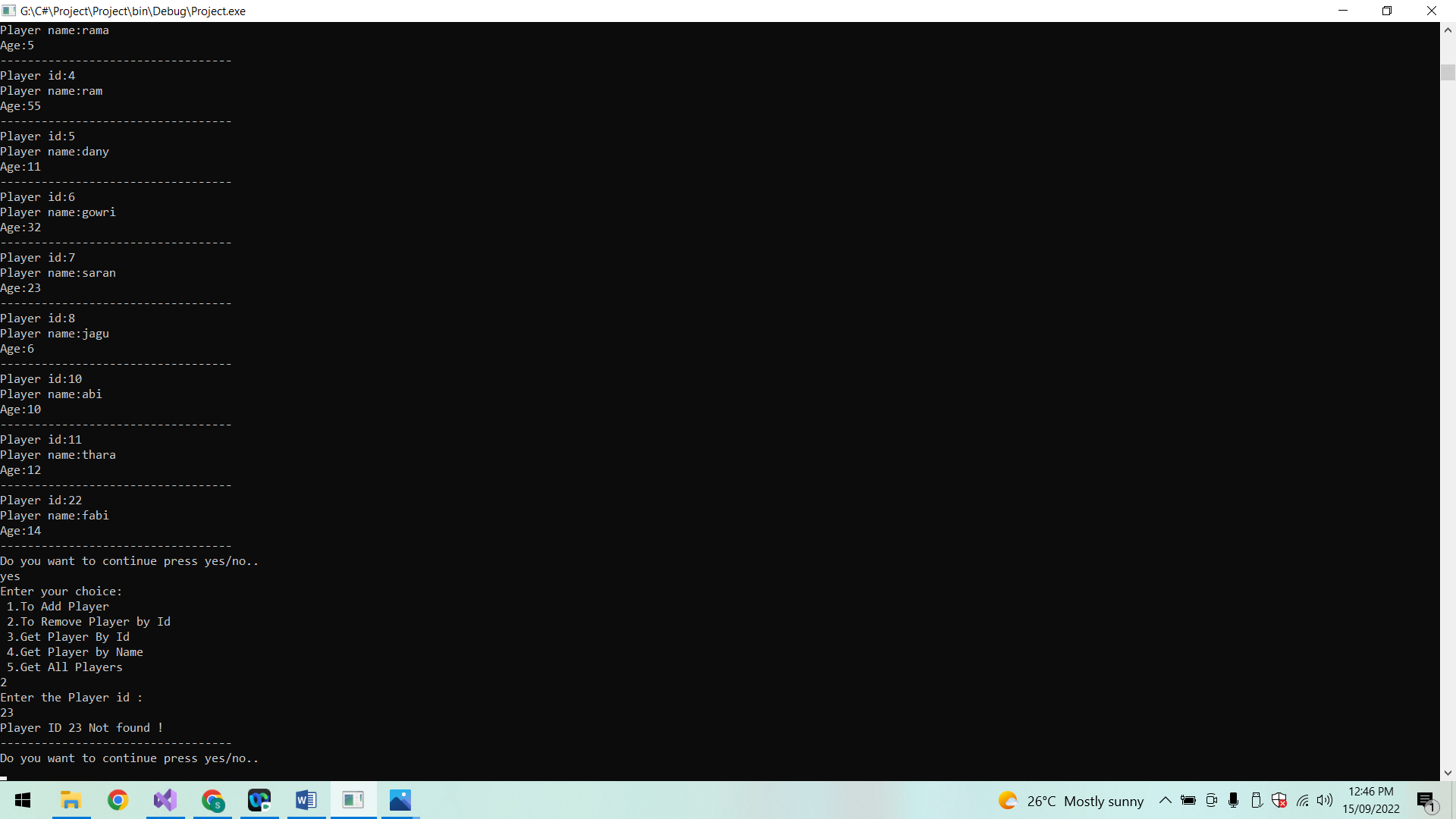
****

**List of players:**

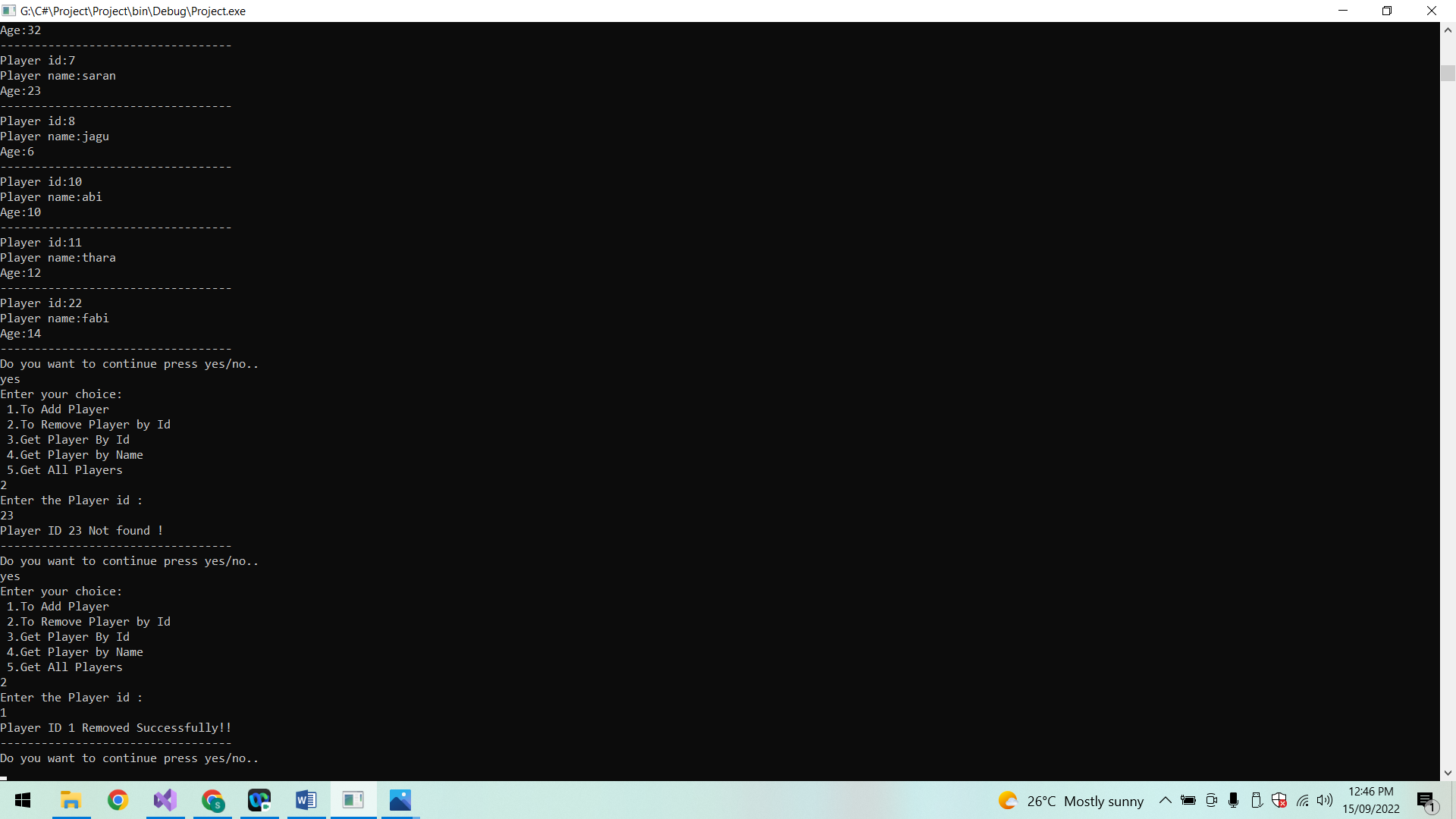
****

**Remove player:**

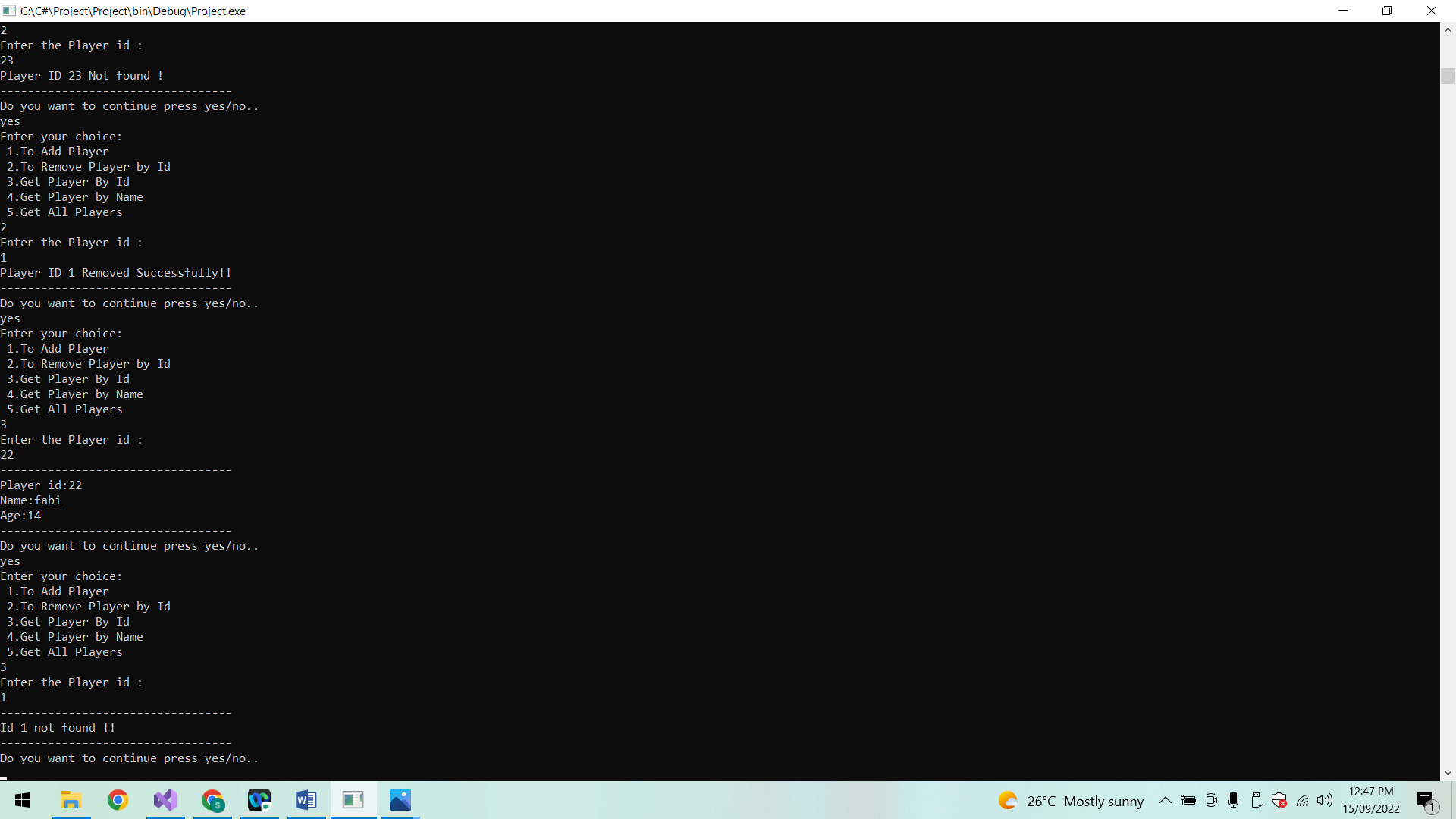
If the player id is not present then display the message “Id not found”.

****

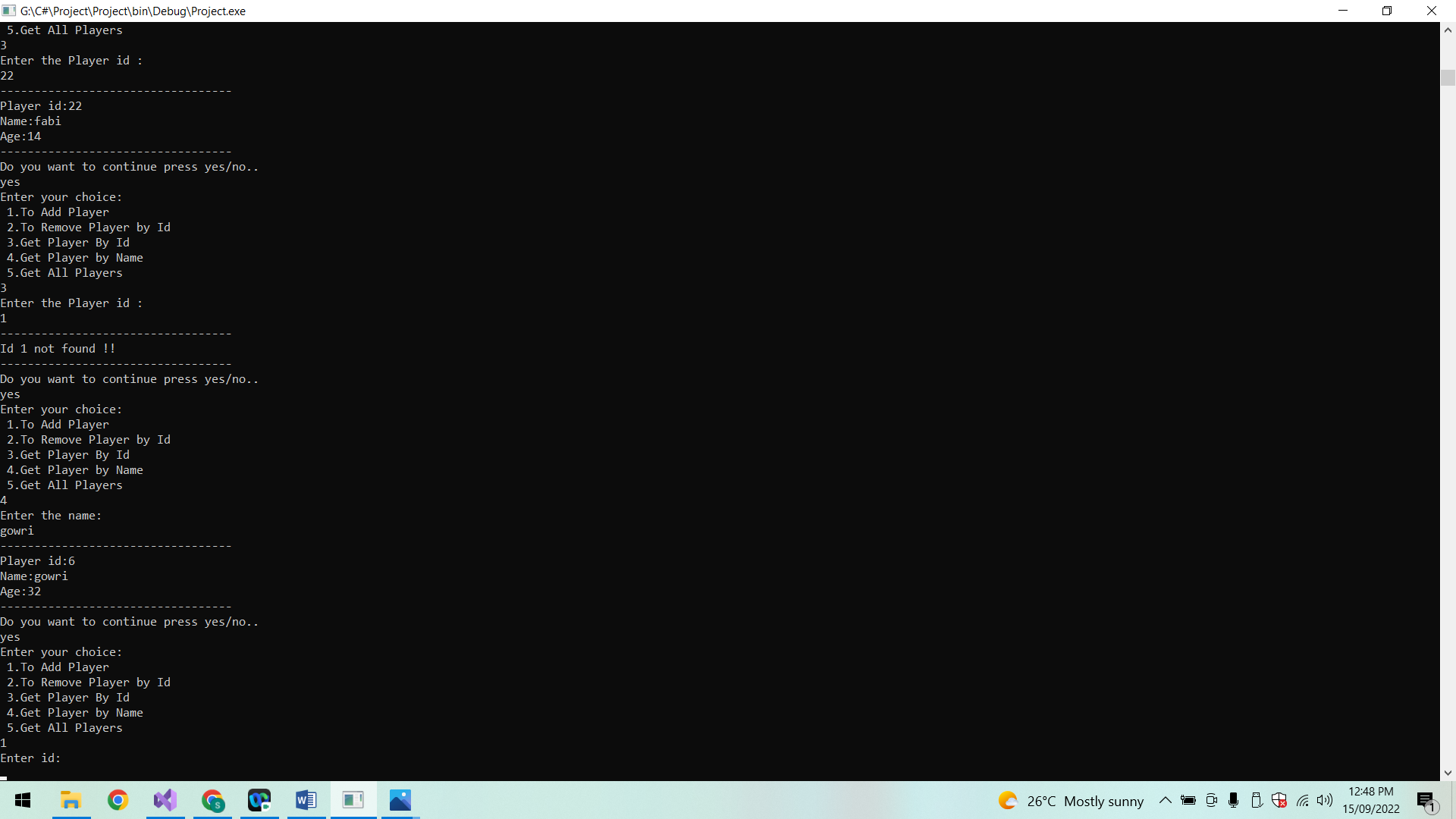
**Remove player:**

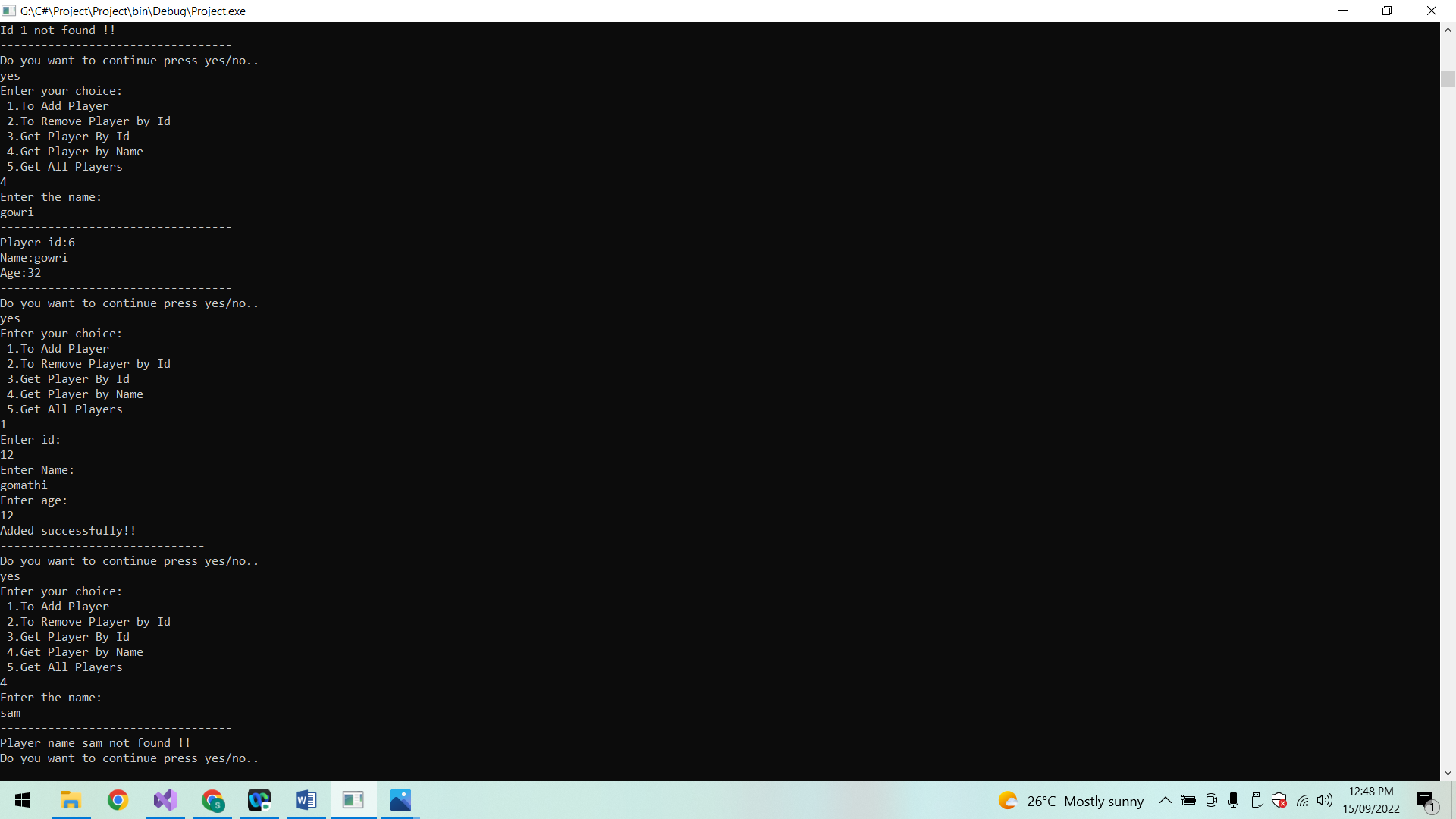
****

**Get player by Id:**

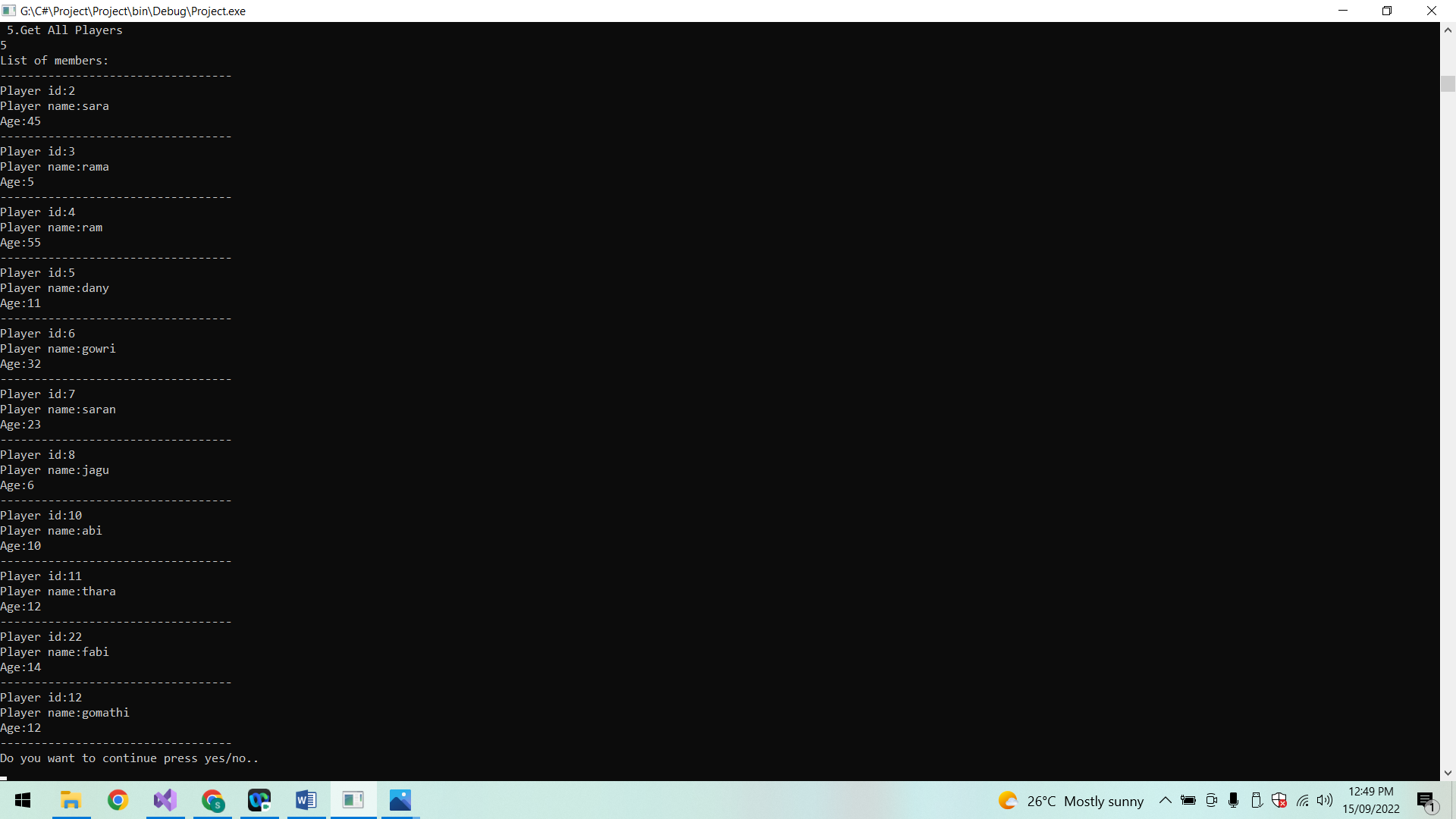
****

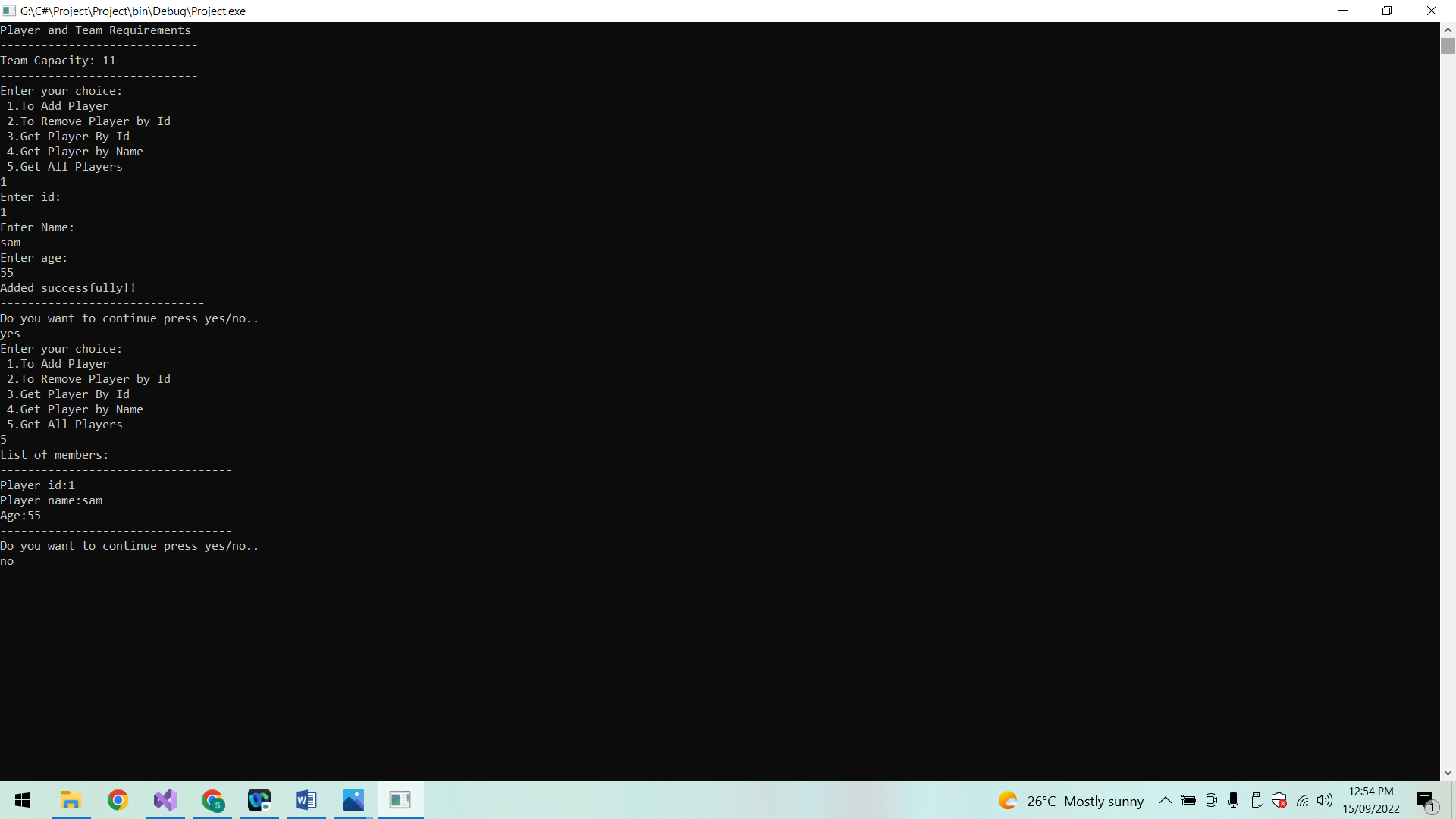
**Get player by name:**

****

****

**List of players:**

****

****

**Conclusion:**

Thus the Player and Team project is implemented successfully.