# Introduction

Welcome to your favorite day of the week which is programming day🎉. This week, we shall work together to learn and implement new programming concepts.

### 

### Let's do some coding.

## 

# University Admission Management System

Students, Let’s create the first version of the UAMS that has two roles

* User
* Admin

Notice the following .gif for better understanding

|  |
| --- |

Lets write down the functions are arrays that will be required for developing this program.

* We need three global arrays for storing name, password, and roles of the users.
* In addition we need a global variable to count the number of students added in these arrays.
* Lastly, we need different single responsbility functions for implementing the logic of UAMS.

|  |
| --- |

Lets define these functions one by one according to their functionality.

* The first menu that will return the choice of the user as an interger number.

|  |
| --- |

* The signIn function that will return the role of the user and take two input parameters as name and password.

|  |
| --- |

* The signUp function that will return a boolean value after taking in input parameters name, password and role of the user.

|  |
| --- |

* topHeader function will be used to print the main header of the UAMS.

|  |
| --- |

* The subMenuBeforeMainMenu will print the submenu

|  |
| --- |

* The subMenu will print the submenu

|  |
| --- |

* clearScreen function will clear the screen.

|  |
| --- |

* adminMenu will print the admin panel on the console.

|  |
| --- |

* adminInterface will be the controller function for admin panel

|  |
| --- |

* userMenu will print the user menu on the console screen.

|  |
| --- |

* userInterface is the driver function for the user panel

|  |
| --- |

Let’s now implement the Main() driver function.

|  |
| --- |

**Congratulations !! You have implemented your first version of the UAMS.**

**Project Version 01:** Complete your business application with the help from Teaching Assitant assigned to you.

# Bonus Tasks (optional)

**Task 01(CP):**

Create a function that determines whether elements in an array can be re-arranged to form a consecutive list of numbers where each number appears exactly once.

**Test Cases:**

| Array Length: 5  [5, 1, 4, 3, 2] | **True** | // Can be re-arranged to form [1, 2, 3, 4, 5] |
| --- | --- | --- |
| Array Length: 6  [5, 1, 4, 3, 2, 8] | **False** |  |
| Array Length: 6  [5, 6, 7, 8, 9, 9] | **False** | // 9 appears twice |

**Task 02(CP):**

Given a list of directions to spin, "left" or "right", return an integer of how many full 360° rotations were made. Note that each word in the array counts as a 90° rotation in that direction.

| ["right", "right", "right", "right", "left", "right"] | 1 | // You spun right 4 times (90 \* 4 = 360)  // You spun left once (360 - 90 = 270)  // But you spun right once more to make a full rotation (270 + 90 = 360) |
| --- | --- | --- |
| ["left", "right", "left", "right"] | 0 |  |
| ["right", "right", "right", "right", "right", "right", "right", "right"] | 2 |  |
| ["left", "left", "left", "left"] | 1 |  |

**Good Luck and Best Wishes !!**

**Happy Coding ahead :)**