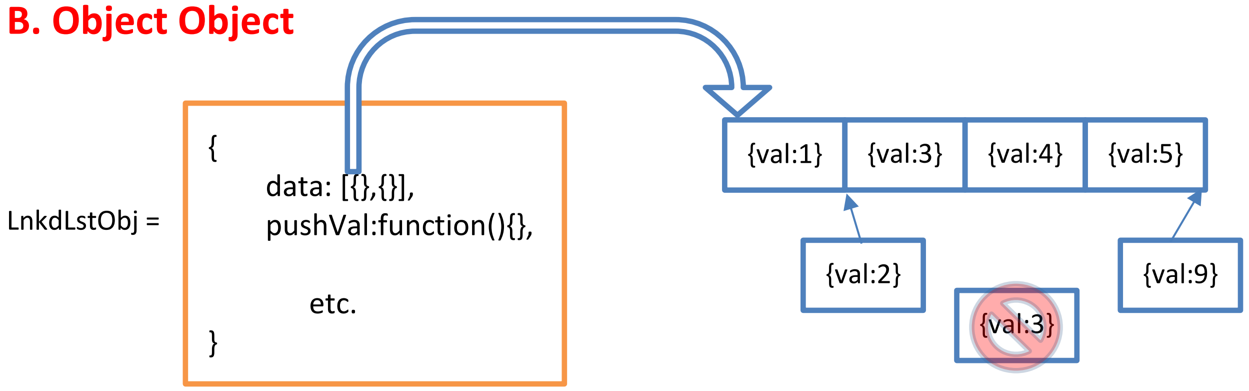
A.Error Object

A.1. Update your cookie.js library file to handle any possible wrong call of all implemented function by firing error message. e.g there should be an error message if getCookie was called without passing any parameter.

B. Object Object



B.1. (BONUS) Make your own simulation for a simple linked list custom Object that accepts objects with a single numeric property value in an ascending order. Let your object has the following functionalities

- **Enqueue a value as long as the value is in the sequence otherwise through an exception (push an item at the end of the list with the passed value).**
- **Insert an item in specific place as long as the value is missing from the sequence otherwise through an exception.**
- **Pop a value (remove an item from the end of the list).**
- **Remove an item from specific place with the required value, if the value is not added return a message with "data not found".**
- **Dequeue a value (remove an item from the beginning of the list).**
- **Display the content of the list.**
- **Ensure that there is no duplication in your entered values.**
- **You can add any property you need.**

Note: You can use Array Object methods and there is no need to use sort() function.

B.2. Using Constructor method for creating Objects, write a script that allows you to create a rectangle object that

- **Should have width and height properties.**
- **Implement two methods for calculating its area and perimeter.**
- **Implement displayInfo() function to display a message declaring the width, height, area and perimeter of the created object.**

C. Function Object

C.1. Create a function that accepts only 2 parameters and throw exception if number of parameters either less than or exceeds 2 parameters

C.2. Create an adding function that adds n numbers only. Throw exception if the user passed any data type other than "number" or called your function without passing any parameters.

C.3. Write two different function with two different forms implementations that takes any number of parameters and returns them reversed using array's reverse function.

C.4. Create your own custom object that has getSetGen as function value, this function should generate setters and getters for the properties of the caller object
This object may have description property of string value if needed
Let any other created object can use this function property to generate getters and setters for his own properties
Avoid generating getters or setters for property of function value

Hint:
if getSetGen() applied on any other object it should generate getters and setters for all of the applied object properties

i.e. if you have the following object
obj = {id:”SD-10”,location:”SV”, addr:”123 st.”, getSetGen: function(){/*should be implemented*/}}
using of getSetGen() will generate the following getId(), setId(), getLocation(), setLocation(), getAddr(), setAddr().

If you created the following object
var user = { name:”Ali”,age:10}
When applying getSetGen() on user object (you can use call or bind or apply), it will result in creating the following : getName(), getAge(),setName(),setAge().

You should make your own interface if needed for the all of the above tasks.
