1. Write a function that takes two numbers as parameter and returns an array containing all prime numbers between those two numbers.

2. Write a function that takes an object of numbers as parameter and returns the an array containing absolute value of those numbers. For example { 1:10, 2:-3, 3:4, a:50 } should return [100, 9, 16, 50]

3. Write a function that takes an array as parameter and returns the sum of the numbers in that array. Array may contain non number properties.

4. Write a function that takes an array as parameter converts it into an object and returns the object. For example: ["a","c","100","d"] will return {1:"a",2:"c",3:"100",4:"d"}

5. Write a function that takes a string as parameter and returns the number of vowels(**a, e, i o, u**) in that string.

6. Write a function that takes two arrays as parameter and merge them into a single array and return. For example: [1,5,4] and [2,100,80] will return [1,5,4,2,100,80]

7. Write a function that takes an integer as parameter and prints out the fibonacci series upto that place. For example is 7 is given as parameter then the output should be 0,1,1,2,3,5,8

8. Write a function that takes an object as parameter and cascadingly prints out the name of all the properties of the object. For example:

```
{
     id: 1,
     firstName:"Adnan",
     lastName:"Nuruddin",
     address:{
          road: 135,
          house: "186/B",
          thana: "Badda",
          district: "Dhaka"
     },
     dateOfBirth:"1-1-2000"
}
```

   The object above should print out: id, firstName, lastName, address, road, house, thana, district, dateOfBirth.

9. Make a function that takes an object as parameter and validates the properties of that object. The following validation needs to be checked:

   Student Object:

   a. Id: integer, required
   b. firstName: string, required
   c. lastName: string, required
   d. phone: numeric, required
   e. classes: array of classes objects, minimum one class required, **check classes object below for further validation**
   f. address: object, **check address object below for further validation**

g. subjectId: array of integers, can be blank, required
h. dateOfBirth: date string, required

Classes Object:

a. classId: integer, required
b. subjectName: string, required
c. obtainedMark: number, minimum 0, maximum 100, not required

Address Object:

a. line1: string, required
b. line2: string
c. thana: string, required
d. district: string, required
e. postCode: numeric, required

If any validation check fails throw an exception explaining the reason of failure and the name of the property.