**EXPERIMENT 1**

**1.a) Write a JavaScript program which accepts a string as input and swap the case of each character. For example if you input 'The Quick Brown Fox' the output should be 'tHEqUICKbROWNfOX'. – “one.js”**

const readline = require('readline');

var RL = readline.createInterface(process.stdin, process.stdout);

RL.question('Please Enter Text: ', (name)=>{

let x=name;

let y="";

for(let i=0;i<x.length;i++)

{

if (x.charAt(i) >='A' && x.charAt(i) <= 'Z')

y=y+x.charAt(i).toLowerCase();

else if(x.charAt(i) >='a' && x.charAt(i) <= 'z')

y=y+x.charAt(i).toUpperCase();

}

console.log(`Output is is ${y}`);

});

**EXPECTED OUTPUT:**

**Sample input: The Quick Brown Fox**

**Sample Output: tHEqUICKbROWNfOX**

b) **. Write a JavaScript program to find the most frequent item of an array. – “two.js”**

var arr1=[3, 'a', 'a', 'a', 2, 3, 'a', 3, 'a', 2, 4, 9, 3];

var mf = 1;

var m = 0;

var item;

for (var i=0; i<arr1.length-1; i++)

{

for (var j=i; j<arr1.length; j++)

{

if (arr1[i] == arr1[j])

m++;

if (mf<m)

{

mf=m;

item = arr1[i];

}

}

m=0;

}

console.log(item+" ( " +mf +" times ) ") ;

**EXPECTED OUTPUT**

**a ( 5 times )**

**if the input contains same number of occurrences for two entries then it prints only least significant one**

**c). Write a JavaScript program to remove duplicate items from an array – ‘three.js’**

function removeDuplicates(num) {

len=num.length;

uniqueChars=[];

num.forEach((c) => {

if (!uniqueChars.includes(c)) {

uniqueChars.push(c);

}

});

return uniqueChars;

}

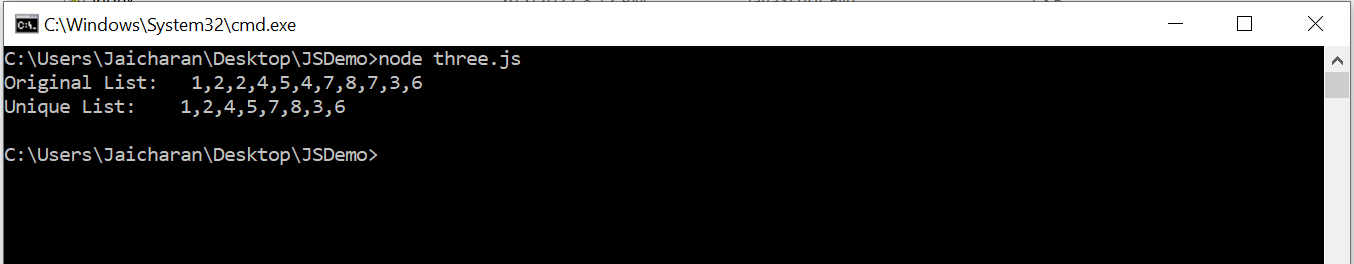
let Mynum = [1, 2, 2, 4, 5, 4, 7, 8, 7, 3, 6];

result = removeDuplicates(Mynum);

console.log("Original List: "+Mynum);

console.log("Unique List: "+result);

**EXPECTED OUTPUT:**

****

**d) Write a JavaScript program to perform a binary search. – “four.js”**

let iterativeFunction = function (arr, x) {

let start=0, end=arr.length-1;

while (start<=end){

let mid=Math.floor((start + end)/2);

if (arr[mid]===x) return true;

else if (arr[mid] < x)

start = mid + 1;

else

end = mid - 1;

}

return false;

}

let arr = [1, 3, 5, 7, 8, 9];

let x = 5;

console.log(iterativeFunction(arr, x) ) ;

**EXPECTED OUTPUT:**

**-> if x : 5 output: true**

**-> if x: 6 output: false**



e) **Write a JavaScript program to list the properties of a JavaScript object – “five.js”**

let object = {

name: 'Jack',

age: 25,

college: 'KMIT',

year: 3,

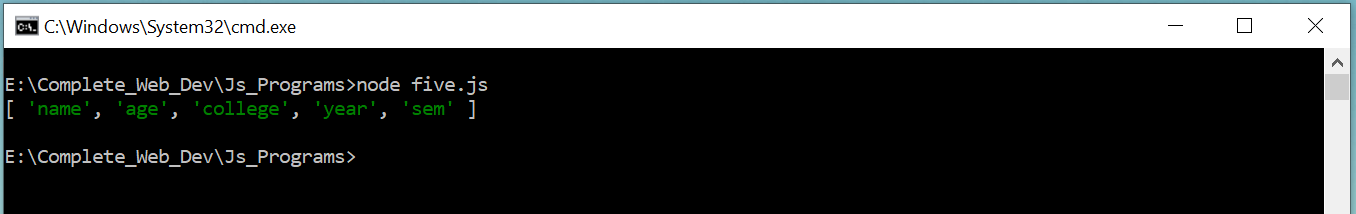
sem: 1

};

let properties = Object.keys(object)

console.log(properties);

**EXPECTED OUTPUT:**



**f) Write a JavaScript function to check whether an object contains given property. –“six.js”**

***1. hasOwnProperty()* method**

let object = {

name: 'Jack',

age: 25,

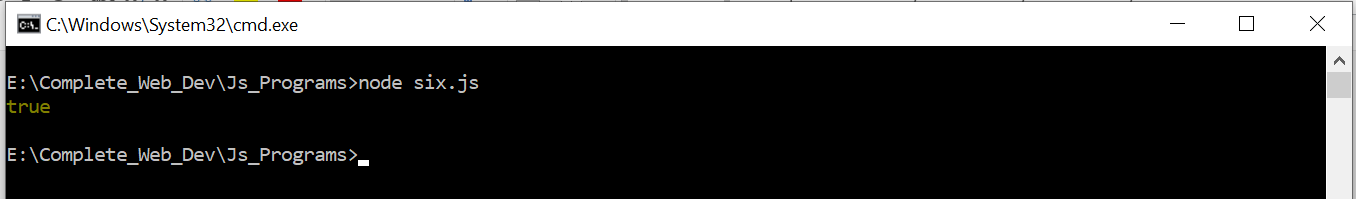
college: 'KMIT',

year: 3,

sem: 1

};

console.log(object.hasOwnProperty('name'));



***2. in operator Method:***

let object = {

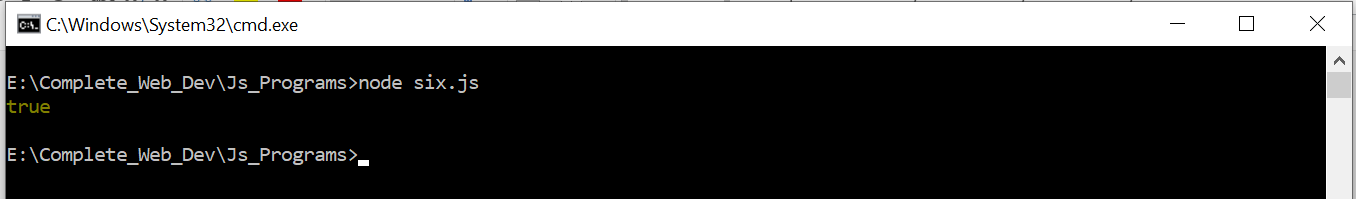
name: 'Jack',

age: 25,

college: 'KMIT', year: 3, sem: 1

};

console.log(‘name’ in object);



***3. Comparing with undefined Method:***

let object = {

name: 'Jack',

age: 25,

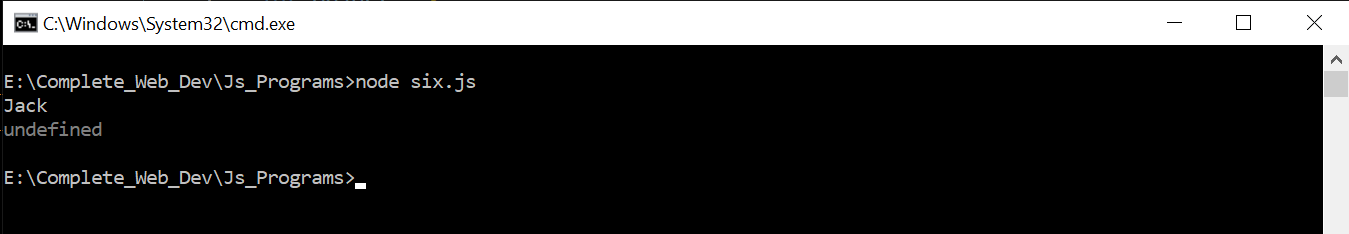
college: 'KMIT', year: 3, sem: 1

};

console.log(object.name);

console.log(object.fee);

Expected output: here Name property is available so programs gives you output as ‘Jack’ but fee property is not available so it is giving output as undefined.



1. **Write a JavaScript program to sort a list of elements using Quick sort. –“seven.js”**

function quick\_Sort(origArray) {

if (origArray.length <= 1) {

return origArray;

} else {

var left = [];

var right = [];

var newArray = [];

var pivot = origArray.pop();

var length = origArray.length;

for (var i = 0; i < length; i++) {

if (origArray[i] <= pivot) {

left.push(origArray[i]);

} else {

right.push(origArray[i]);

}

}

return newArray.concat(quick\_Sort(left), pivot, quick\_Sort(right));

}

}

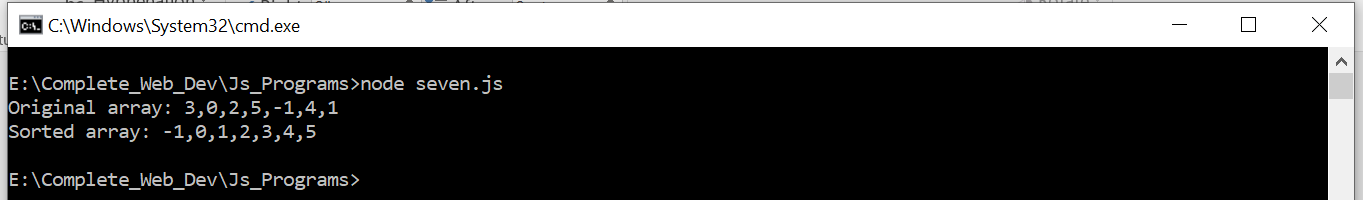
var myArray = [3, 0, 2, 5, -1, 4, 1 ];

console.log("Original array: " + myArray);

var sortedArray = quick\_Sort(myArray);

console.log("Sorted array: " + sortedArray);

**Expected Output:**



1. **Write a JavaScript program to implement Bubble Sort. –“eight.js”**

function swap(arr, first\_Index, second\_Index){

var temp = arr[first\_Index];

arr[first\_Index] = arr[second\_Index];

arr[second\_Index] = temp;

}

function bubble\_Sort(arr){

var len = arr.length,

i, j, stop;

for (i=0; i < len; i++){

for (j=0, stop=len-i; j < stop; j++){

if (arr[j] > arr[j+1]){

swap(arr, j, j+1);

}

}

}

return arr;

}

myArray=[3, 0, 2, 5, -1, 4, 1];

console.log("Original array: " + myArray);

var sortedArray = bubble\_Sort(myArray);

console.log("Sorted array: " + sortedArray);

Expected Output:

