question 4   
  
var now =new Date();  
var day = now.getDay();  
var obj ={  
0:"Monday",  
1:"Tuesday",  
2:"Wednesday",  
3:"Thursday",  
4:"Friday",  
5:"Saturday",  
6:"Sunday",  
};  
console.log("Today is :",obj[day]);  
var hours=now.getHours();  
var minutes=now.getMinutes();  
var seconds=now.getSeconds();  
console.log("current time :",hours,":",minutes,":",seconds);  
  
//question 5  
  
var time = now.getDate();  
var month = now.getMonth();  
var year = now.getFullYear();  
console.log(time,"-",month+1,"-",year);\*/  
  
/\*  
// question 1   
// add function add() - 0  
function add(){  
return 0  
}  
//add(1,2) - 3  
function add(x,y){  
return 3  
}  
let x = 1;  
let y = 2;  
let sum = x+y;  
console.log("sum is :",sum);  
  
// question 2   
function add(a, mycode) {  
return a + mycode;  
}  
  
let a = 1;  
let mycode = "one";  
let sum2 = add(a, mycode);  
console.log("sum is:", sum2); // Output: sum is: 1one  
  
// question 3   
function add(...args) {  
let sum = 0;  
args.forEach((arg) => {  
if (typeof arg === "number") {  
sum += arg;  
} else if (Array.isArray(arg)) {  
// If it's an array, recursively call the add function to sum its elements  
sum += add(...arg);  
} else if (typeof arg === "string" && !isNaN(Number(arg))) {  
// If it's a numeric string, convert it to a number and add to the sum  
sum += Number(arg);  
}  
// If it's not a number or array, ignore it (e.g., non-numeric string)  
});  
return sum;  
}  
  
let result = add(1, [2, 3, "4"], "1", "one");  
console.log("Result is:", result); // Output: Result is: 11  
  
//question 4   
let one = 1;  
let two = 2;  
  
function add(...args) {  
return args.reduce((sum, arg) => {  
if (!isNaN(arg) && arg !== "") {  
return sum + Number(arg);  
} else if (Array.isArray(arg)) {  
return sum + add(...arg);  
} else {  
return sum;  
}  
}, 0);  
}  
  
let result = add(1, [one, two, 3, "4"], one, two, "one", "two");  
console.log("Result is:", result); // Output: Result is: 14   
  
// question 2  
const one = () => 1;  
const two = () => "2";  
  
function add(...args) {  
return args.reduce((sum, arg) => {  
if (!isNaN(arg) && arg !== "") {  
return sum + Number(arg);  
} else if (Array.isArray(arg)) {  
return sum + add(...arg);  
} else if (typeof arg === "function") {  
return sum + Number(arg());  
} else {  
return sum;  
}  
}, 0);  
}  
  
console.log("add():", add()); // Output: add(): 0  
console.log("add(1,2):", add(1, 2)); // Output: add(1,2): 3  
console.log("add(1,\"2\"):", add(1, "2")); // Output: add(1,"2"): 3  
console.log("add(1,\"one\"):", add(1, "one")); // Output: add(1,"one"): 0  
console.log("add(1,[2,3,\"4\"],\"1\",\"one\"):", add(1, [2, 3, "4"], "1", "one")); // Output: add(1,[2,3,"4"],"1","one"): 11  
console.log("add(1,[one, two, 3, \"4\"], one, two, \"one\", \"two\"):", add(1, [one, two, 3, "4"], one, two, "one", "two")); // Output: add(1,[one, two,3, "4"], one, two, "one", "two"): 14  
  
// question 3   
/\*The spread operator is used to split an array or an object into individual elements or properties. It is commonly used for array literals or when calling functions that expect multiple arguments.  
The rest operator is used to collect multiple arguments into an array. It allows you to handle an arbitrary number of arguments more conveniently.  
\*/  
// spread array example  
const arr1 = [1, 2, 3];  
const arr2 = [...arr1, 4, 5];  
  
console.log(arr2); // Output: [1, 2, 3, 4, 5]  
  
// spread object example  
const obj1 = { a: 1, b: 2 };  
const obj2 = { ...obj1, c: 3 };  
console.log(obj2); // Output: { a: 1, b: 2, c: 3 }