**Problem 1 :**

|  |
| --- |
| const getProperties=(obj)=>{  //1-first solution    return Object.keys(obj);   //2-second Solution   return Object.getOwnPropertyNames(obj);  }  const b = {      a:"hi",      b:"bye",      c:"hello"  }  console.log(getProperties(b)); |

**Problem 2:**

|  |
| --- |
| **const isPlainObject=(o)=>{**  **/\* console.log(Object.prototype.toString.call(o))**  **var c= Object.prototype.toString.call(o) == '[object Object]'**  **&& o.constructor && o.constructor.name == "Object";**  **return c === true;\*/**  **return typeof(o)==='object' && o !== null && !Array.isArray(o)**  **}**  **console.log( isPlainObject({"a":1,"b":2}));**  **console.log( isPlainObject({}));**  **console.log( isPlainObject(null));**  **console.log( isPlainObject([1,2]));** |

**Problem 3:**

|  |
| --- |
| **const isPlainObject = (Obj) => {**  **return (Obj.constructor.name === 'Object')**  **}**  **const getPropertiesNb = (obj) => {**  **return Object.keys(obj).length;**  **}**  **const modifyObject = (obj) => {**  **obj.isObject = isPlainObject(obj);**  **obj.getPropertiesNb = getPropertiesNb(obj);**  **return obj;**  **}**  **console.log(modifyObject({ a: 'Mostafa' }));**  **console.log(modifyObject([1, 2, 5]));** |

**Problem 4:**

|  |
| --- |
| **const makePairs = (obj) => {**  **return JSON.stringify(Object.entries(obj));**  **}**  **console.log(makePairs({ a: 1, c: 2 }))** |

**Problem 5:**

|  |
| --- |
| **const without = (object, ...name) => {**  **//console.log(name);**  **name.forEach((element) => delete object[element]);**  **return object;**  **};**  **var obj = {**  **a: 1,**  **b: 2,**  **};**  **console.log(without(obj, "a", "b"));** |

**Problem 6:**

|  |
| --- |
| **const isEmpty = (obj) => {**  **if (obj === null || Object.keys(obj).length === 0) {**  **return true;**  **}**  **else {**  **return Object.keys(obj).filter(key => {**  **if (obj[key] === false || obj[key] === null || obj[key] === undefined) {**  **return false;**  **}**  **return true;**  **}).length === 0**  **}**  **}**  **console.log(isEmpty({}));**  **console.log(isEmpty({ a: undefined }));**  **console.log(isEmpty({ a: 1 }));**  **console.log(isEmpty({ a: 1, b: undefined }))** |

**Problem 7:**

|  |
| --- |
| **const isEqual =(obj1,obj2)=>{**  **return (JSON.stringify(obj1) === JSON.stringify(obj2))**  **}**  **console.log(isEqual ({ a: 1, b: 1 }, { a: 1, b: 2 } ));**  **console.log(isEqual ({ a: 1, b: 1 }, { a: 1, b: 1 } ));**  **const isEqual2 =(obj1,obj2)=>{**  **const keys1=Object.keys(obj1);**  **const keys2=Object.keys(obj2);**  **if (keys1.length !== keys2.length) {**  **return false**  **}**  **for(let key of keys1){**  **if( obj1[key]!==obj2[key] ){**  **return false**  **}**    **}** |