**TASK2**

**Q2) Write a write up on difference between copy by value and copy by reference?**

**CALL BY VALUE :**

1. While calling a function, when you pass values by copying variables, it is known as "Call By Values."
2. In this method, a copy of the variable is passed.
3. Changes made in a copy of variable never modify the value of variable outside the function
4. Does not allow you to make any changes in the actual variables.
5. Values of variables are passed using a straightforward method
6. Actual and formal arguments will be created in different memory location
7. Default in many programming languages like C++.PHP. Visual Basic NET, and C#.
8. Actual arguments remain safe as they cannot be modified accidentally.

**CALL BY REFERENCE:**

1. While calling a function, in programming language instead of copying the values of variables, the address of the variables is used it is known as "Call By References.
2. In this method, a variable itself is passed.
3. Change in the variable also affects the value of the variable outside the function.
4. Allows you to make changes in the values of variables by using function calls.
5. Pointer variables are required to store the address of variables.
6. It is supported by most programming languages like JAVA, but not as default.
7. Actual and formal arguments will be created in the same memory location.
8. Actual arguments are not Safe. They can be accidentally modified, so you need to handle arguments operations carefully.

**Q3) How to copy by value a composite data type [array + objects]?**

***COPY BY VALUE A COMPOSITE DATA TYPE***

String,Number,Boolean are primitive data types whereas Objects and Arrays are composite data types in JavaScript.

primitive data types are passed in by value

composite data types i.e.., non-primitive data types are passed in by reference

let’s see the basic program of copy by value

**Let a =2;**

**Let b=a;**

**console.log(b,a)//2,2**

**console.log(b,a)//2,4**

Imagine,these variables of data used to store our values declaring a variable a and similarly the value of a is stored in b. We can allocate another variable c and now we have also changed the value of c only the original value is copied and it does not disturb the value of c hence, we pass the primitive value directly. value do not affect the original value.

Similarly, let’s see the basic program of copy by reference

**Let copybyref={**

**a :2**

**}**

**Let b={…3};**

**console.log(copybyref.a)//2**

**copybyref.a=5;**

**console.log(copybyref.a)//5**

In this example, When we create a variable javascript stores it’s location. when we copy the variable we copy the location of our original value is copied i.e.., copybyref.a is referencing the location of a.If we change any value of it,and it will also change the original value.

**How do we copy by value in a composite data type ?**

The most intermediate way to approach copy by value by using spread operator.

**Let arr = [1,2,3];**

**Let arr2 = […arr];**

**console.log(arr); // [ 1,2,3 ]**

**arr2.push(‘d’); //**inserting an element at the end of arr2

**console.log(arr2); // [ 1,2,3,4 ]**

**console.log(arr);**

Here, arr 2 has the elements of arr1 copied into it if any changes are made it would reflect the other, In this way the spread operator is used .