**Task 2**

**2.Copy by value vs Copy by Reference?**

To understand the nuanced difference between the two types of copy we need to understand what happens in the memory after we assign a value to it

COPY BY VALUE:

In a primitive data type when a variable is assigned avalue we can imagine that a box is created in the memory.This box has a sticker attached to it I.e the variable name.Inside the box the value assigned to the variable is stored.

var x=17

var y=’xyz’

x contains a value 17 and y contains a value ‘xyz’

Now if,

Var a=x;

Var b=y;

The values in the boxes x and y are copied into the variables a and b.Even though x and a and y and b contains the same value they are not connected to each other.It is so because the values are directly copied into the new variables.Changes taking place in one does not affect the other

COPY BY REFERENCE:

In case of non primitive data types the values are not directly copied.When a non primitive data type is assigned a value of the box is createdwith a sticker of the name of the dta type.However the values it is assigned is not stored directly in the box.The language itself assigns a different memory location to store the data.The address of this memory locationis stored in the box created

Let user={name:’Ram’};

Let admin=user;

admin.name=’Shyam’;

when the value of the admin is changed it automatically changesthe value of the user as well.This happens because both the user and admin are storing the address of the memory location and when one changes the values in the allocated memory it is reflected in the other as well.

**3.How to copy by value a composite datatypes?**

Using JSON.parse() and JSON.stringify():

The JSON object,available in all modern browsers,has two useful methods to deal with JSON-formatted content:parse and stringify.JSON.parse() takes a JSON string and transforms it into a JS object.JSON.stringify() takes a JS object and transforms it into a JSON string

a=[1,2,3]

var b=JSON.parse(JSON.stringify(a))

This performs a deep copy changes in one doesn’t affect the other.