TASK-2 DATA TYPES

1. x=5;

y=x;

x=10;

console.log(x);

console.log(y);

ans: Output**: 10**

5

The variable `x` is initially assigned the value `5`. Then, `y` is assigned the current value of `x`, which is `5`. When `x` is updated to `10`, `y` remains `5`, so the output will be `10` for `x` and `5` for `y`.

2. obj1 = {name: "allice"};

obj2 = obj1;

obj1.name = "bob";

console.log(obj1.name);

console.log(obj2.name);

ans: output bob

bob

`obj1` is initially set with the property `name` equal to `"allice"`. When `obj2` is assigned `obj1`, both variables reference the same object. Updating `obj1.name` to `"bob"` changes the shared object, so the output will be `"bob"` for both `obj1.name` and `obj2.name`.

3. a ="hello";

b = 42;

c = true;

d = {key: "value"};

e = null;

f = undefined;

```
console.log(typeof a);

console.log(typeof b);

console.log(typeof c);

console.log(typeof d);

console.log(typeof e);

console.log(typeof f);
```

ans: string
number
boolean
object
object
undefined

The `typeof` operator returns the data type of each variable. For `a`, it outputs `"string"`; for `b`, it outputs `"number"`; for `c`, it outputs `"boolean"`; for `d`, it outputs `"object"`; for `e`, it outputs `"object"` (since `null` is a special case); and for `f`, it outputs `"undefined"`. Thus, the console will display: `"string"`, `"number"`, `"boolean"`, `"object"`, `"object"`, and `"undefined"`.

4.
```
numbers=[10,20,30,40,50];

console.log(numbers[2]);

console.log(numbers[0]);

console.log(numbers[numbers.length-1]);
```

ans: output 30
10
50
The array numbers contains five elements: [10, 20, 30, 40, 50]. The expression numbers[2] accesses the third element, which is 30, while numbers[0] retrieves the first element, 10. Finally, numbers[numbers.length-1] accesses the last element of the array, which is 50

5.
```
fruits =["apple","banana","mango"];
fruits [1]= "orange";
console.log(fruits);
```
ans: output ['apple', 'orange', 'mango']

The code initializes an array called fruits containing three items: "apple," "banana," and "mango." It then updates the second item (index 1) from "banana" to "orange." The console.log(fruits)

6.matrix = [
[1,2,3],
[4,5,6],
[7,8,9]
];
console.log(matrix[1][2]);
console.log(matrix[2][0]);

Output**: 6**
7

The code defines a 2D array (matrix) with three rows and three columns. matrix[1][2] accesses the element in the second row and third column, which is 6. matrix[2][0] accesses the element in the third row and first column, which is 7

7. person={
name:'THARUN',
age:'22',
city:'khammam'
};
console.log(person.name);
console.log(person.age);

Output: **THARUN**
22

The code defines an object person with properties: name, age, and city. console.log(person.name) retrieves the value of the name property, which is 'THARUN'. console.log(person.age) retrieves the value of the age property, which is '22'

8.car ={
make:'toyota',
model:'corolla',
year:'2021'
};
console.log(car["make"]);
console.log(car["model"]);
Output**: toyota**
corolla

The code defines an object car with properties: make, model, and year. console.log(car["make"]) retrieves the value of the make property, which is 'toyota'. console.log(car["model"]) retrieves the value of the model property, which is 'corolla'

9. book={
title:"the great gatsby",
author:"F.scott fitzgerald"
};
book.author="anonymous";
console.log(book.author);

output**: anonymous**

The code defines an object book with properties title and author. The author property is then updated to 'anonymous'. When console.log(book.author) is executed,

```
10. student={
    name:"Tharun",
    grade:"A",
};
student.age="20";
console.log(student);
```

output: **{ name: 'Tharun', grade: 'A', age: '20' }**

The code defines an object student with properties name and grade. A new property age is added and set to '20'. When console.log(student) is executed