**Expt\_7:Set operations**

<html>

<head>

<title>Set Operation</title>

</head>

<body>

<script>

// Union

function union(setA, setB) {

let unionSet = new Set(setA);

for (let elem of setB) {

unionSet.add(elem);

}

return unionSet;

}

// Intersection

function intersection(setA, setB) {

let intersectionSet = new Set();

for (let elem of setB){

if (setA.has(elem)) {

intersectionSet.add(elem);

}

}

return intersectionSet;

}

//Difference

function difference(setA,setB) {

let differenceSet = new Set(setA);

for (let elem of setB) {

differenceSet.delete(elem);

}

return differenceSet;

}

// Symmetric Difference

function symmetricDifference(setA,setB){

let differenceSet = new Set(setA);

for (let elem of setB) {

if(differenceSet.has(elem)){

differenceSet.delete(elem);

}

else {

differenceSet.add(elem);

}

}

return differenceSet;

}

// Example sets

const setA = new Set(['apple', 'banana', 'cherry']);

const setB = new Set(['banana', 'date', 'fig']);

// Display results using document.write

document.write('Union:'+ Array.from(union(setA, setB)).join(',')+'<br>');

document.write('Intersection:'+ Array.from(intersection(setA, setB)).join(',')+'<br>');

document.write('Difference:'+ Array.from(difference(setA, setB)).join(',')+'<br>');

document.write('Symmetric Difference:'+ Array.from(symmetricDifference(setA, setB)).join(',')+'<br>');

</script>

</body>

</html>