```
//-------------------------------------------------------------------
-------------------------------------------------------------------
-------
// 1. W.A.P with using array methods like push(), pop(), unshift(),
shift(), toString(),
// join(), concat(), at(), length, indexOf().
// (Inputs=>const
city=['pune’,’delhi’,’mumbai’,’bengaluru’,’mysuru’])

//ANS:------

// const cities = ['pune', 'delhi', 'mumbai', 'bengaluru',
'mysuru'];

// // push()
// cities.push('chennai');
// console.log(cities);

// // pop()
// const poppedCity = cities.pop();
// console.log(poppedCity);

// // unshift()
// cities.unshift('kolkata');
// console.log(cities);

// // shift()
// const shiftedCity = cities.shift();
// console.log(shiftedCity);

// // toString()
// const citiesString = cities.toString();
// console.log(citiesString);

// // join()
// const joinedCities = cities.join(' | ');
// console.log(joinedCities);

// // concat()
// const moreCities = ['hyderabad', 'ahmedabad'];
// const combinedCities = cities.concat(moreCities);
// console.log(combinedCities);
```

```
// // at() - using bracket notation
// const cityAtIndex2 = cities[2];
// console.log(cityAtIndex2);

// // length
// const numberOfCities = cities.length;
// console.log(numberOfCities);

// // indexOf()
// const indexMumbai = cities.indexOf('mumbai');
// console.log(indexMumbai);

//-------------------------------------------------------------------
-------------------------------------------------------------------
-------
// 2. W.A.P with using splice() in array (cond: remove the first
three elements and
// replace with new three new elements )
// (Inputs=>const
city=['pune’,’delhi’,’mumbai’,’bengaluru’,’mysuru’])

//ANS:-----

// const cities = ['pune', 'delhi', 'mumbai', 'bengaluru',
'mysuru'];

// // Using splice() to remove the first three elements and replace
them
// cities.splice(0, 3, 'Indore', 'Ujjain', 'Barwani');

// // Display the modified array
// console.log(cities);

//-------------------------------------------------------------------
-------------------------------------------------------------------
--------
// 3. W.A.P with using slice() in array (cond: only display the
“delhi” ,”mumbai”)
// (Inputs=>const
city=['pune’,’delhi’,’mumbai’,’bengaluru’,’mysuru’])
```

```
//ANS:-----------------------------

// const cities = ['pune', 'delhi', 'mumbai', 'bengaluru',
'mysuru'];

// // Using slice() to extract "delhi" and "mumbai"
// const selectedCities = cities.slice(1, 3);

// // Display the selected cities
// console.log(selectedCities);

//-----------------------------------------------------------------
-----------------------------------------------------------------
--------
// 4. W.A.P with using strings methods like includes(), charAt(),
concat(),
// toUpperCase, toLowercase.
//  (Inputs=>const flowers=”Sunflower, mariegold, rose, Lotus”)

//ANS:----

// const flowers = "Sunflower, marigold, rose, Lotus";

// // Using includes() to check if a string contains another string
// const includesRose = flowers.includes("rose");
// console.log("Includes 'rose':", includesRose);

// // Using charAt() to get the character at a specific position
// const charAtPosition5 = flowers.charAt(5);
// console.log("Character at position 5:", charAtPosition5);

// // Using concat() to concatenate two strings
// const moreFlowers = ", daisy, tulip";
// const combinedFlowers = flowers.concat(moreFlowers);
// console.log("Combined flowers:", combinedFlowers);

// // Using toUpperCase() to convert the string to uppercase
// const uppercasedFlowers = flowers.toUpperCase();
// console.log("Uppercased flowers:", uppercasedFlowers);

// // Using toLowerCase() to convert the string to lowercase
// const lowercasedFlowers = flowers.toLowerCase();
```

```
// console.log("Lowercased flowers:", lowercasedFlowers);

//-------------------------------------------------------------------
-------------------------------------------------------------------
-------
// 5. W.A.P with using slice() in strings (cond: take the parameters
like
// 1.(5,13), 2.(6), 3. (-9), 4.(-7,-4) )
//  (Inputs=>const flowers=”Sunflower, mariegold, rose, Lotus”)

//ANS:------------------

// const flowers = "Sunflower, marigold, rose, Lotus";

// // Using slice(5, 13) to extract a substring from index 5 to
index 12 (13-1)
// const substring1 = flowers.slice(5, 13);
// console.log("Substring 1:", substring1);

// // Using slice(6) to extract a substring starting from index 6 to
the end
// const substring2 = flowers.slice(6);
// console.log("Substring 2:", substring2);

// // Using slice(-9) to extract the last 9 characters from the
string
// const substring3 = flowers.slice(-9);
// console.log("Substring 3:", substring3);

// // Using slice(-7, -4) to extract a substring from the 7th
character from the end to the 4th character from the end
// const substring4 = flowers.slice(-7, -4);
// console.log("Substring 4:", substring4);

//-------------------------------------------------------------------
-------------------------------------------------------------------
-------
// 6. W.A.P with using substring() and substr() in strings
//  (Inputs=>const flowers=”Sunflower, mariegold, rose, Lotus”)

//ANS: --------------
```

```
// const flowers = "Sunflower, marigold, rose, Lotus";

// // Using substring(5, 13) to extract a substring from index 5 to
index 12 (13-1)
// const substring1 = flowers.substring(5, 13);
// console.log("Substring 1:", substring1);

// // Using substring(6) to extract a substring starting from index
6 to the end
// const substring2 = flowers.substring(6);
// console.log("Substring 2:", substring2);

// // Using substr(9) to extract a substring starting from index 9
to the end
// const substring3 = flowers.substr(9);
// console.log("Substring 3:", substring3);

// // Using substr(-5) to extract the last 5 characters from the
string
// const substring4 = flowers.substr(-5);
// console.log("Substring 4:", substring4);

//-----------------------------------------------------------------
-----------------------------------------------------------------
-------
// 7. W.A.P with using replace() and replaceAll() in strings
//  (inputs=>const text=”I love fruits. fruits are very tasty.
fruits are very healthy”)

//ANS:-----------------------

// const text = "I love fruits. fruits are very tasty. fruits are
very healthy";

// // Using replace() to replace the first occurrence of "fruits"
with "apples"
// const replacedText = text.replace("fruits", "apples");
// console.log("Replaced Text (replace()):", replacedText);

// // Using replaceAll() to replace all occurrences of "fruits" with
"apples"
// const replacedAllText = text.replaceAll("fruits", "apples");
```

```
// console.log("Replaced Text (replaceAll()):", replacedAllText);

//-----------------------------------------------------------------------
-----------------------------------------------------------------------
--------
// 8. W.A.P with using indexOf() and lastIndexOf() in strings
//  (inputs=> const text=”javascript is super, javascript is
awesome”)

//ANS:-------------------

// const text = "javascript is super, javascript is awesome";

// // Using indexOf() to find the first occurrence of "javascript"
// const indexOfJavascript = text.indexOf("super");
// console.log("Index of 'javascript' (indexOf()):",
indexOfJavascript);

// // Using lastIndexOf() to find the last occurrence of
"javascript"
// const lastIndexOfJavascript = text.lastIndexOf("is");
// console.log("Last Index of 'javascript' (lastIndexOf()):",
lastIndexOfJavascript);

//-----------------------------------------------------------------------
-----------------------------------------------------------------------
--------
// 9. W.A.P to print multiline statements using template literals.

//ANS:-----------------

// Multiline statement using template literals
// const multilineStatement = `
//   This is a multiline statement using template literals.
//   It allows you to easily create strings that span multiple
lines.
//   Template literals use backticks (\`) instead of single or
double quotes.
//   You can also include expressions like ${2 + 2} directly in the
string.
//   `;
// console.log(multilineStatement);
```

```
//-----------------------------------------------------------------------
-----------------------------------------------------------------------
--------
```