Sure, I can provide you with examples of these programs using anonymous functions and IIFE (Immediately Invoked Function Expressions) in JavaScript. Here are the programs:

1. Print odd numbers in an array using an anonymous function:
```javascript
const arr = [1, 2, 3, 4, 5, 6, 7, 8, 9];
arr.forEach(function(num) {
  if (num % 2 !== 0) {
    console.log(num);
  }
});
```

2. Convert all the strings to title caps in a string array using an anonymous function:
```javascript
const strArray = ["hello", "world", "javascript"];
const titleCapsArray = strArray.map(function(str) {
  return str.charAt(0).toUpperCase() + str.slice(1);
});
console.log(titleCapsArray);
```

3. Sum of all numbers in an array using an anonymous function:
```javascript
const numbers = [1, 2, 3, 4, 5];
const sum = numbers.reduce(function(acc, num) {
  return acc + num;
}, 0);
console.log(sum);
```

4. Return all the prime numbers in an array using an anonymous function:
```javascript
function isPrime(num) {
  if (num <= 1) return false;
  for (let i = 2; i <= Math.sqrt(num); i++) {
    if (num % i === 0) {
      return false;
    }
  }
  return true;
}

const arr = [2, 3, 4, 5, 6, 7, 8, 9];
```

```
const primeNumbers = arr.filter(function(num) {
  return isPrime(num);
});
console.log(primeNumbers);
```

5. Return all the palindromes in an array using an anonymous function:
```javascript
function isPalindrome(str) {
  const reversed = str.split("").reverse().join("");
  return str === reversed;
}

const strArray = ["racecar", "hello", "level", "world"];
const palindromes = strArray.filter(function(str) {
  return isPalindrome(str);
});
console.log(palindromes);
```

6. Return the median of two sorted arrays of the same size using an anonymous function:
```javascript
function findMedianSortedArrays(arr1, arr2) {
  const mergedArray = arr1.concat(arr2).sort((a, b) => a - b);
  const mid = Math.floor(mergedArray.length / 2);
  if (mergedArray.length % 2 === 0) {
    return (mergedArray[mid - 1] + mergedArray[mid]) / 2;
  } else {
    return mergedArray[mid];
  }
}

const array1 = [1, 3, 8];
const array2 = [2, 4, 9];
const median = findMedianSortedArrays(array1, array2);
console.log(median);
```

7. Remove duplicates from an array using an anonymous function:
```javascript
const arr = [1, 2, 2, 3, 4, 4, 5];
const uniqueArray = arr.filter(function(value, index, self) {
  return self.indexOf(value) === index;
});
```

```
console.log(uniqueArray);
```

8. Rotate an array by k times using an anonymous function:
```javascript
function rotateArray(arr, k) {
  const n = arr.length;
  k = k % n;
  return arr.slice(k).concat(arr.slice(0, k));
}

const originalArray = [1, 2, 3, 4, 5];
const k = 2;
const rotatedArray = rotateArray(originalArray, k);
console.log(rotatedArray);
```

IIFE can also be used to encapsulate these functions if needed.

Certainly! Here are the programs you mentioned using arrow functions in JavaScript:

1. Print odd numbers in an array using an arrow function:
```javascript
const arr = [1, 2, 3, 4, 5, 6, 7, 8, 9];
arr.forEach(num => {
  if (num % 2 !== 0) {
    console.log(num);
  }
});
```

2. Convert all the strings to title caps in a string array using an arrow function:
```javascript
const strArray = ["hello", "world", "javascript"];
const titleCapsArray = strArray.map(str => str.charAt(0).toUpperCase() + str.slice(1));
console.log(titleCapsArray);
```

3. Sum of all numbers in an array using an arrow function:
```javascript
const numbers = [1, 2, 3, 4, 5];
const sum = numbers.reduce((acc, num) => acc + num, 0);
```

```
console.log(sum);
```

4. Return all the prime numbers in an array using an arrow function:
```javascript
const isPrime = num => {
  if (num <= 1) return false;
  for (let i = 2; i <= Math.sqrt(num); i++) {
    if (num % i === 0) {
      return false;
    }
  }
  return true;
};

const arr = [2, 3, 4, 5, 6, 7, 8, 9];
const primeNumbers = arr.filter(num => isPrime(num));
console.log(primeNumbers);
```

5. Return all the palindromes in an array using an arrow function:
```javascript
const isPalindrome = str => {
  const reversed = str.split("").reverse().join("");
  return str === reversed;
};

const strArray = ["racecar", "hello", "level", "world"];
const palindromes = strArray.filter(str => isPalindrome(str));
console.log(palindromes);
```

Arrow functions provide a more concise way to define functions, especially when they are simple and don't require a lot of additional logic.