1)Print odd numbers in an array

anonymous function

var odd=function(arr)

{

var arr1="";

for(var i=0;i<arr.length;i++)

{

if((arr[i] % 2) !== 0)

{

arr1=arr1+" "+arr[i]+ " ";

}

}

console.log(arr1);

}

odd([1,2,3,4,5,6,7,8,9]);

Output:

1 3 5 7 9

Execution Time:

0.065s

Memory Used:

8084kb

IIFE FUNCTION

(function(arr){

var arr1="";

for(var i=0;i<arr.length;i++)

{

if((arr[i] % 2) !== 0)

{

arr1=arr1+" "+arr[i]+ " ";

}

}

console.log(arr1);

})([1,2,3,4,5,6,7,8,9]);

Output:

Output:

1 3 5 7 9

Execution Time:

0.065s

Memory Used:

8144kb

2)Convert all the strings to title caps in a string array

anonymous function

var titlecaps=function(arr)

{

var arr1="";

for(var i=0;i<arr.length;i++)

{

arr1= arr1+" "+arr[i].charAt(0).toUpperCase()+arr[i].slice(1,arr[i].length);

}

console.log(arr1);

}

titlecaps(["vicky","divya","kirthik"])

Output:

Vicky Divya Kirthik

Execution Time:

0.065s

Memory Used:

8120kb

IIFE FUNCTION

(function(arr)

{

var arr1="";

for(var i=0;i<arr.length;i++)

{

arr1= arr1+" "+arr[i].charAt(0).toUpperCase()+arr[i].slice(1,arr[i].length);

}

console.log(arr1);

})(["vicky","divya","kirthik"]);

Output:

Vicky Divya Kirthik

Execution Time:

0.065s

Memory Used:

8088kb

3)Sum of all numbers in an array

Anonymous function

var sum=function(arr)

{

var sum=0;

for(var i=0;i<arr.length;i++)

{

sum=sum+arr[i];

}

console.log(sum);

}

sum([1,2,3,4,5]);

Output:

15

Execution Time:

0.069s

Memory Used:

8052kb

IIFE Function

(function(arr)

{

var sum=0;

for(var i=0;i<arr.length;i++)

{

sum=sum+arr[i];

}

console.log(sum);

})([1,2,3,4,5]);

Output:

15

Execution Time:

0.065s

Memory Used:

8036kb

4)Return all the prime numbers in an array

Anonymous function

var prime=function(arr)

{

var arr1="";

var sum=0;

for(var i=0;i<arr.length;i++)

{

for(var j=2;j<=arr[i];j++)

{

if((arr[i] % j) ===0)

{

sum=sum+1;

}

}

if(sum===1)

{

arr1=arr1+" "+arr[i]+" ";

}

sum=0;

}

console.log(arr1);

}

prime([5,4,2,3]);

Output:

5 2 3

Execution Time:

0.064s

Memory Used:

8120kb

IIFE Function

(function(arr)

{

var arr1="";

var sum=0;

for(var i=0;i<arr.length;i++)

{

for(var j=2;j<=arr[i];j++)

{

if((arr[i] % j) ===0)

{

sum=sum+1;

}

}

if(sum===1)

{

arr1=arr1+" "+arr[i]+" ";

}

sum=0;

}

console.log(arr1);

})([5,4,2,3]);

Output:

5 2 3

Execution Time:

0.065s

Memory Used:

8028kb

5)Return all the palindromes in an array

Anonymous function

var palindrome=function(arr)

{

var arr1="";

var arr2="";

for(var i=0;i<arr.length;i++)

{

for(var j=arr[i].length-1;j>=0;j--)

{

arr1= arr1+arr[i].charAt(j);

}

if(arr[i]==arr1)

{

arr2=arr2+" "+arr1;

}

arr1="";

}

console.log(arr2);

}

palindrome(["malayalam","Kirthik","madam","vignesh"])

Output:

malayalam madam

Execution Time:

0.065s

Memory Used:

8128kb

IIFE Function

(function(arr)

{

var arr1="";

var arr2="";

for(var i=0;i<arr.length;i++)

{

for(var j=arr[i].length-1;j>=0;j--)

{

arr1= arr1+arr[i].charAt(j);

}

if(arr[i]==arr1)

{

arr2=arr2+" "+arr1;

}

arr1="";

}

console.log(arr2);

})(["malayalam","Kirthik","madam","vignesh"]);

Output:

malayalam madam

Execution Time:

0.066s

Memory Used:

8072kb

6)Return median of two sorted arrays of same size

Anonymous function

var median=function(arr1,arr2)

{

var temp;

var arr=arr1.concat(arr2);

console.log("Merged array :"+arr);

for(var i=0;i<arr.length;i++)

{

for(var j=i+1;j<arr.length;j++)

{

if(arr[i]>arr[j])

{

temp=arr[i];

arr[i]=arr[j];

arr[j]=temp;

}

}

}

console.log("Sorted array :"+arr);

var n=arr.length;

var n1=n/2-1;

var n2=(arr[n1]+arr[n1+1])/2;

console.log("Median :"+n2);

}

median([1,3,5,7],[2,4,6,8]);

Output:

Merged array :1,3,5,7,2,4,6,8

Sorted array :1,2,3,4,5,6,7,8

Median :4.5

Execution Time:

0.065s

Memory Used:

8056kb

IIFE Function

(function(arr1,arr2)

{

var temp;

var arr=arr1.concat(arr2);

console.log("Merged array :"+arr);

for(var i=0;i<arr.length;i++)

{

for(var j=i+1;j<arr.length;j++)

{

if(arr[i]>arr[j])

{

temp=arr[i];

arr[i]=arr[j];

arr[j]=temp;

}

}

}

console.log("Sorted array :"+arr);

var n=arr.length;

var n1=n/2-1;

var n2=(arr[n1]+arr[n1+1])/2;

console.log("Median :"+n2);

})([1,3,5,7],[2,4,6,8]);

Output:

Merged array :1,3,5,7,2,4,6,8

Sorted array :1,2,3,4,5,6,7,8

Median :4.5

Execution Time:

0.066s

Memory Used:

8112kb

7)Remove duplicates from an array

Anonymous function

var dublicate=function(arr)

{

for(var i=0;i<arr.length;i++)

{

for(j=i+1;j<arr.length;j++)

{

if(arr[i]==arr[j])

{

arr.splice(j,1);

}

}

}

console.log(arr);

}

dublicate([1,2,4,5,1,6,2]);

Output:

[ 1, 2, 4, 5, 6 ]

Execution Time:

0.068s

Memory Used:

8112kb

IIFE Function

(function(arr)

{

for(var i=0;i<arr.length;i++)

{

for(j=i+1;j<arr.length;j++)

{

if(arr[i]==arr[j])

{

arr.splice(j,1);

}

}

}

console.log(arr);

})([1,2,4,5,1,6,2]);

Output:

[ 1, 2, 4, 5, 6 ]

Execution Time:

0.066s

Memory Used:

8112kb

8)Rotate an array by k times

Anonymous function

var rotate=function(arr,k)

{

var arr1= arr.slice(k).concat(arr.slice(0, k));

console.log(arr1);

}

rotate([1,2,3,4,5],2);

Output:

[ 3, 4, 5, 1, 2 ]

Execution Time:

0.066s

Memory Used:

8108kb

IIFE Function

(function(arr,k)

{

var arr1= arr.slice(k).concat(arr.slice(0, k));

console.log(arr1);

})([1,2,3,4,5],3);

Output:

[ 4, 5, 1, 2, 3 ]

Execution Time:

0.066s

Memory Used:

8116kb