JAVASCRIPT FULL STACK DEVELOPER

TASK 3

1) **Print odd numbers in an array**

a) **Anonymous Function:**

Code:

```
const readline = require('readline');
const inp = readline.createInterface({
  input: process.stdin
});
const userInput = [];
inp.on("line", (data) => {
 userInput.push(data);
});
inp.on("close", () => {

var oddnum = function (given)
{
   for (var i in given)
   {
      if(given[i]%2 != 0)
      {
         console.log(given[i])
      }
   }
}
oddnum ([1, 2, 3, 4])
});
```

Output:

```
1
3
```

b) **Arrow Function:**

Code:

```
const readline = require('readline');
const inp = readline.createInterface({
  input: process.stdin
});
const userInput = [];
inp.on("line", (data) => {
 userInput.push(data);
```

```
});
inp.on("close", () => {

var odd = (given) => {

 for (var i in given)
   {
      if(given[i]%2 != 0)
      {
         console.log(given[i])
      }
   }
}
odd ([1,99,98,90,13,16,67])
});
```

Output:

```
1
99
13
67
```

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

a) **Anonymous Function:**

Code:

```
const readline = require('readline');
const inp = readline.createInterface({
  input: process.stdin
});
const userInput = [];
inp.on("line", (data) => {
 userInput.push(data);
});
inp.on("close", () => {

var titleCase = function (str) {
  str = str.toLowerCase().split(' ');
  for (var i = 0; i < str.length; i++) {
    str[i] = str[i].charAt(0).toUpperCase() + str[i].slice(1);
  }
  return str.join(' ');
}
console.log( titleCase("zen class task 3")) ;
});
```

Output:

```
Zen Class Task 3
```

a) **Arrow Function:**

Code:

```
const readline = require('readline');
const inp = readline.createInterface({
  input: process.stdin
});
const userInput = [];
inp.on("line", (data) => {
 userInput.push(data);
});
inp.on("close", () => {

var titleCase = (str) =>
{
  str = str.toLowerCase().split(' ');
  for (var i = 0; i < str.length; i++) {
    str[i] = str[i].charAt(0).toUpperCase() + str[i].slice(1);
  }
  return str.join(' ');
}
console.log( titleCase("zen class task 3")) ;
});
```

Output:

```
Zen Class Task 3
```

3) **Sum of all numbers in an array**

a) **Anonymous Function:**

Code:

```
const readline = require('readline');
const inp = readline.createInterface({
  input: process.stdin
});
const userInput = [];
inp.on("line", (data) => {
 userInput.push(data);
});
inp.on("close", () => {
```

```
var sum = function(a)
{
var total=0;
for ( i in a)
{
total += a[i];
}
return total;
}
console.log(sum([10,20,40]))
});
```

Output:

```
70
```

b) **Arrow Function:**

Code:

```
const readline = require('readline');
const inp = readline.createInterface({
  input: process.stdin
});
const userInput = [];
inp.on("line", (data) => {
 userInput.push(data);
});
inp.on("close", () => {

var odd = (given) => {

 var sum = (a) =>
{
   var total=0;
   for ( i in a)
   {
      total += a[i];
   }
   return total;
}
console.log(sum([100,20,40]));
});
```

Output:

```
160
```

4) **Return all the prime numbers in an array**

a) **Anonymous Function:**

Code:

```
const readline = require('readline');
const inp = readline.createInterface({
  input: process.stdin
});
const userInput = [];
inp.on("line", (data) => {
 userInput.push(data);
});
inp.on("close", () => {
var prime = function(given)
{

   var primearray=[];

  for(var i in given)
   { var x = "yes"
      for (var j=2; j < given[i]; j++)
      {
         if((given[i]%j == 0))
         {

            x = "no"
         }

      }
     if(x=="yes" && given[i]!=1 && given[i]!=0)
     {
        primearray.push(given[i])
     }
   }
   return primearray
}

console.log(prime([0,1, 2,3,4,5,6,7,8]))
});
```

Output:

```
[ 2, 3, 5, 7 ]
```

b) Arrow Function:

Code:

```
const readline = require('readline');
const inp = readline.createInterface({
  input: process.stdin
});
const userInput = [];
inp.on("line", (data) => {
 userInput.push(data);
});
inp.on("close", () => {
var prime = (given) =>
{

   var primearray=[];

  for(var i in given)
   { var x = "yes"
      for (var j=2; j < given[i]; j++)
      {
         if((given[i]%j == 0))
         {

            x = "no"
         }

      }
     if(x=="yes" && given[i]!=1 && given[i]!=0)
     {
        primearray.push(given[i])
     }
   }
   return primearray
}
console.log(prime([7,9,8,4,2,3,1]))
});
```

Output:

```
[ 7, 2, 3 ]
```

5) **Return all the palindromes in an array**

a) **Anonymous Function:**

Code:

```
const readline = require('readline');
const inp = readline.createInterface({
  input: process.stdin
});
const userInput = [];
inp.on("line", (data) => {
 userInput.push(data);
});
inp.on("close", () => {
var palindrome = function(a)
{
for(var i in a)
{
   var x= a[i].toString().split("").reverse().join("")
   if(x==a[i])
   {
      console.log(a[i]);
   }

}
}
palindrome(["malayalam" , 99 , "mom", "happy", 5, 198])
});
```

Output:

```
malayalam
99
mom
5
```

b) **Arrow Function:**

Code:

```
const readline = require('readline');
const inp = readline.createInterface({
  input: process.stdin
});
const userInput = [];
```

```
inp.on("line", (data) => {
 userInput.push(data);
});
inp.on("close", () => {
var palindrome = (a) =>
{
for(var i in a)
{
   var x= a[i].toString().split("").reverse().join("")
   if(x==a[i])
   {
      console.log(a[i]);
   }
}
}
palindrome(["malayalam" , 99 , "mom", "happy", 909, 198])
});
```

Output:

```
malayalam
99
mom
909
```

6) **Return median of two sorted arrays of same size**

a) **Anonymous Function:**

Code:

```
const readline = require('readline');
const inp = readline.createInterface({
  input: process.stdin
});
const userInput = [];
inp.on("line", (data) => {
 userInput.push(data);
});
inp.on("close", () => {
var median = function(a,b)
{
a = a.sort(function(a,b){return(a-b)});
b = b.sort(function(a,b){return(a-b)});
c=a.concat(b).sort(function(a,b){return(a-b)})
var median = ((c[((a.length)-1)]+c[(a.length)] ) / 2)
```

```
median =(median.toFixed(2));

return median;
}
console.log(median([0,22,89,86] , [11,13,16,77]))
});
```

Output:

```
19.00
```

b) Arrow Function:

Code:

```
const readline = require('readline');
const inp = readline.createInterface({
  input: process.stdin
});
const userInput = [];
inp.on("line", (data) => {
 userInput.push(data);
});
inp.on("close", () => {
var median = (a,b) =>
{
a = a.sort(function(a,b){return(a-b)});
b = b.sort(function(a,b){return(a-b)});
c=a.concat(b).sort(function(a,b){return(a-b)})
var median = ((c[((a.length)-1)]+c[(a.length)] ) / 2)
median =(median.toFixed(2));

return median;
}
console.log(median([0,22,89,86] , [11,13,16,77]))
});
```

Output:

```
19.00
```

7) **Remove duplicates from an array**

a) **Anonymous Function:**

Code:

```
const readline = require('readline');
const inp = readline.createInterface({
  input: process.stdin
});
const userInput = [];
inp.on("line", (data) => {
 userInput.push(data);
});
inp.on("close", () => {
var nodup = function(a)
{
   var b = a.slice(0,a.length)
   var c=[];
for (var i in a)
{
  var count=0;
   for( var j in b)
   {
      if(a[i]==b[j])
      {
         count+=1;
         delete b[j]

         if(count==1)
      {
         c.push(a[i])
      }

      }
   }
}
return c ;
}
console.log(nodup ([1,1,1,1,2,2,2,3,4,5,6,6,7]))
});
```

Output:

```
[ 1, 2, 3, 4, 5, 6, 7 ]
```

b) Arrow Function:

Code:

```
const readline = require('readline');
const inp = readline.createInterface({
  input: process.stdin
});
const userInput = [];
inp.on("line", (data) => {
 userInput.push(data);
});
inp.on("close", () => {
var nodup = (a) =>
{
   var b = a.slice(0,a.length)
   var c=[];
for (var i in a)
{
  var count=0;
   for( var j in b)
   {
      if(a[i]==b[j])
      {
         count+=1;
         delete b[j]

         if(count==1)
      {
         c.push(a[i])
      }
      }
   }
}
return c ;
}
console.log(nodup ([1,1,1,1,"apple","apple"]))});
```

Output:

```
[ 1, 'apple' ]
```

8) **Rotate an array by k times and return the rotated array**

a) **Anonymous Function:**

Code:

```
const readline = require('readline');
const inp = readline.createInterface({
  input: process.stdin
});
const userInput = [];
inp.on("line", (data) => {
 userInput.push(data);
});
inp.on("close", () => {
var rotate = function(a,k)
{
   for(var i=1 ; i<=k ; i++)
   {
      a.unshift(a[(a.length)-1])
      a.pop();
  //console.log(a)
   }
   return a;
}
console.log(rotate([1,2,4,5,6],2))
)};
```

Output:

```
[ 5, 6, 1, 2, 4 ]
```

b) **Arrow Function:**

Code:

```
const readline = require('readline');
const inp = readline.createInterface({
  input: process.stdin
});
const userInput = [];
inp.on("line", (data) => {
 userInput.push(data);
});
inp.on("close", () => {
```

```
var rotate = (a,k) =>
{
   for(var i=1 ; i<=k ; i++)
   {
      a.unshift(a[(a.length)-1])
      a.pop();
    //console.log(a)
   }
   return a;
}
console.log(rotate([1,2,5,6],2))
)};
```

Output:

```
[ 5, 6, 1, 2 ]
```