1. Write a typescript program which contains one function named as Maximum. That function accepts
three parameters and it should returns largest value from three input parameters

```
/**
* Function will return maximum from three numbers
* @param {number} no1
* @param {number} no2
* @param {number} no3
*//**
 * Function will return maximum from three numbers
 * @param {number} no1
 * @param {number} no2
 * @param {number} no3
 */
function maximum(no1:number, no2:number, no3: number): number {
    return (no1>no2) ? ((no1>no3) ?/**
 * Function will return maximum from three numbers
 * @param {number} no1
 * @param {number} no2
 * @param {number} no3
 */
function maximum(no1:number, no2:number, no3: number): number {
    return (no1>no2) ? ((no1>no3) ? no1:((no2>no3) ? no2 : no3)):(no2>no3?no2:no3);
}

var max = maximum(23, 89, 6);
console.log("Maximum number is ",max); no1:((no2>no3) ? no2 : no3)):(no2>no3?
no2:no3);
}

var max = maximum(23, 89, 6);
console.log("Maximum number is ",max);
function maximum(no1:number, no2:number, no3: number): number {
return (no1>no2) ? ((no1>no3) ? no1:((no2>no3) ? no2 : no3)):(no2>no3?no2:no3);
}

var max = maximum(23, 89, 6);
console.log("Maximum number is ",max);
```

---

2. Write a typescript program which contains one function named as Area. That function should calculate area of circle. Accept value of radius from user and return its area. Default value of PI should be 3.14 if it is not provided by the caller

```
/**
* Function to calculate area of circle
* @param {number} radius
* @param {number} pi
*/
function areaOfCircle(radius:number, pi:number = 3.14):number {
```

```
return pi * radius * radius;
}

var area = areaOfCircle(5);
console.log("Area of circle: "+area);
```

---

3. Write a typescript program which contains one function named as DisplayFactors. That function should accept one number and display factors of that number.

```
/**
* Function will return factors of number
* @param number
*/
function getFactors(number:number):number[]{
var factors:number[]=[];
for(var i=1; i<number/2+1; i++){
if(number%i==0){
factors.push(i);
}
}
return factors;
}

var factor = getFactors(20);

for(var i=0;i<factor.length;i++){
console.log(factor[i]+ " ");
}
```

---

4. Write a typescript program which contains one function named as ChkPrime. That function should
accept one number and it should return true if the given number is prime and otherwise return false.

```
/**
* Function will check number is prime or not
* @param {number} no
*/
function isPrime(no:number):boolean {
for(var i=2; no>=square(i);i++){
if(no%i==0){
return false;
}
}
return true;
}

/**
* Function will return square of number
```

```
* @param {number} no
*/
function square(no:number):number {
return no*no;
}

if(isPrime(11)){
console.log("Number is prime");
}
else {
console.log("Number is not prime")
}
```

---

5. Write a typescript program which contains one function named as Fibonacci. That function accept
one number from user and print Fibonacci series till that number

```
/**
* Function will return fibonacci series
* @param {number} no
*/
function fibonacci(no:number):number[]{
var fibonacciSeries:number[]=[];
var f1:number = 1;
var f2:number = 1;
var f3:number;
f3=f1+f2;
if(no==f2){
fibonacciSeries.push(f1);
return fibonacciSeries;
}
else{
fibonacciSeries.push(f1);
fibonacciSeries.push(f2);
for(var i = 0; no>=f3; f3 = f1+f2){
fibonacciSeries.push(f3);
f1=f2;
f2=f3;
}
return fibonacciSeries;
}
}

var fibSeries = fibonacci(21);
for(var i=0;i<fibSeries.length;i++){
console.log(fibSeries[i]);
}
```