SCRIPTING LAB ASSIGNMENT

Creation of Calculator App in Angular

1. app.component.css

```
input{
 padding: 10px;
 margin: 10px;
 }
```

2. app.component.html

```
<h1 style="text-align: center;">CALCULATOR</h1>
 <div style="text-align: center;">
   <input placeholder="NUM1" #num1>
   <input placeholder="NUM2" #num2>
</div>

 <div style="text-align: center;" >
 <app-addition [num1]="num1.value" [num2]="num2.value" (sendAddition)="getResult($ev
ent)" style="display: inline-block;"></app-addition>
 <app-subtraction [num1]="num1.value" [num2]="num2.value" (sendSubtraction)="getResu
lt($event)" style="display: inline-block;"></app-subtraction>
 <app-multiplication [num1]="num1.value" [num2]="num2.value" (sendMultiplication)="g
etResult($event)" style="display: inline-block;"></app-multiplication>
 <app-division [num1]="num1.value" [num2]="num2.value" (sendDivision)="getResult($ev
ent)" style="display: inline-block;"></app-division>
</div>

<h2 style="text-align: center;">RESULT={{res1}}</h2>
<div style="text-align: center;">
  <input placeholder="NUM3" #num3>
</div>

<div style="text-align: center;">
<app-factorial [num3]="num3.value" (sendFactorial)="getResult2($event)" style="disp
lay:inline-block;"></app-factorial>
<app-prime [num3]="num3.value" (sendPrime)="getResult2($event)" style="display:inli
ne-block;"></app-prime>
</div>

<div [ngSwitch]="res2">
 <h2 style="text□align: center;" *ngSwitchCase="0">Check Prime:Not a Prime!</h2>
 <h2 style="text□align: center;" *ngSwitchCase="1">Check Prime:Prime Number!</h2>
 <h2 style="text-align: center;" *ngSwitchDefault>Factorial:{{res2}}</h2>
</div>

<ng-template #other_content>other content here...</ng-template>
```

3. app.component.ts

```
import { Component, Input } from '@angular/core';

@Component({
 selector: 'app-root',
 templateUrl: './app.component.html',
 styleUrls: ['./app.component.css']
})
export class AppComponent {
 title = 'calculator';

 @Input() num1!: number;
 @Input() num2!: number;
 @Input() num3!: number;

 res1!: number;
 res2!:number;

 getResult(result:number)
 {
 this.res1=result;
 }

 getResult2(result:number)
 {
 this.res2=result;
 }
}
```

4. addition.component.css

```
button{
 padding: 25px;
 margin: 20px;
 font-size: 40px;
 border-radius: 20px;
 border: 100px;
 border-color: black;
 background-color: aquamarine;
 }
```

5. addition.component.html

```
<button (click)="add(num1,num2)">+</button>
```

6. addition.component.ts

```
import { Component, Input, Output, EventEmitter } from '@angular/core';

@Component({
 selector: 'app-addition',
 templateUrl: './addition.component.html',
 styleUrls: ['./addition.component.css']
})

export class AdditionComponent{

 @Input() num1!:string;
 @Input() num2!:string;

 res!:number;

 @Output() sendAddition = new EventEmitter<number>();

 add(first: string, second: string)
 {
 this.res= parseInt(first) + parseInt(second);
 this.sendAddition.emit(this.res);
 }
}
```

7. division.component.css

```
  button{
  padding: 25px;
  margin: 20px;
  font-size: 40px;
  border-radius: 20px;
  border: 100px;
  border-color: black;
  background-color: aqua;
  }
```

8. division.component.html

```
  <button (click)="division(num1,num2)">/</button>
```

9. division.component.ts

```ts
import { Component, EventEmitter, Input, OnInit, Output } from '@angular/core' ;

@Component({
 selector: 'app-division',
 templateUrl: './division.component.html',
 styleUrls: ['./division.component.css']
})

export class DivisionComponent {

 @Input() num1!:string;
 @Input() num2!:string;

 res!:number;

 @Output() sendDivision = new EventEmitter<number>();

 division(first: string, second: string)
 {
 this.res= parseInt(first) / parseInt(second);
 this.sendDivision.emit(this.res);
 }
}
```

10. factorial.component.css

```css
button{
  padding: 25px;
  margin: 20px;
  font-size: 40px;
  border-radius: 20px;
  border: 100px;
  border-color: black;
  background-color: charteuse;
 }
```

11. factorial.component.html

```
<button (click)="factorial(num3)">Factorial!</button>
```

12. factorial.component.ts

```
import { Component, EventEmitter, Input, Output } from '@angular/core';

@Component({
 selector: 'app-factorial',
 templateUrl: './factorial.component.html',
 styleUrls: ['./factorial.component.css']
})

export class FactorialComponent {

 @Input() num3!:string;

 res!:number;

 @Output() sendFactorial = new EventEmitter<number>();

 fact(n:number):number {
 if (n === 0) {
 return 1;
 }

 return n * this.fact(n - 1);
 }
```

13. multiplication.component.css

```
button{
 padding: 25px;
 margin: 20px;
 font-size: 40px;
 border-radius: 20px;
 border: 100px;
 border-color: black;
 background-color: coral;
}
```

14. multiplication.component.html

```
<button (click)="product(num1,num2)">x</button>
```

15. multiplication.component.ts

```
import { Component, Input, EventEmitter, Output } from '@angular/core';

@Component({
 selector: 'app-multiplication',
 templateUrl: './multiplication.component.html',
 styleUrls: ['./multiplication.component.css']
})

export class MultiplicationComponent{

 @Input() num1!:string;
 @Input() num2!:string;

 res!:number;

 @Output() sendMultiplication = new EventEmitter<number>();

 product(first: string, second: string)
 {
 this.res= parseInt(first) * parseInt(second);
 this.sendMultiplication.emit(this.res);
 }
}
```

16. prime.component.css

```css
button{
 padding: 25px;
 margin: 20px;
 font-size: 40px;
 border-radius: 20px;
 border: 100px;
 border-color: black;
 background-color: charteuse;
}
```

17. prime.component.html

```html
<button (click)="prime(num3)">Check Prime</button>
```

18. prime.component.ts

```
import { Component,EventEmitter, Input, Output} from '@angular/core';

@Component({
 selector: 'app-prime',
 templateUrl: './prime.component.html',
 styleUrls: ['./prime.component.css']
})

export class PrimeComponent {
 @Input() num3!:string;
 res!:number;
 @Output() sendPrime = new EventEmitter<number>();

 check(n:number)
 {
 let isPrime = true;

 for (let i = 2; i < n; i++) {
 if (n % i == 0) {
 isPrime = false;
 break;
 }
 }

 if (isPrime) {
 return 1;
 } else {
 return 0;
 }
 }

 prime(num: string)
 {
 this.res= this.check(parseInt(num));
 this.sendPrime.emit(this.res);
 }
}
```

19. subtraction.component.css

```css
button{
 padding: 25px;
 margin: 20px;
 font-size: 40px;
 border-radius: 20px;
 border: 100px;
 border-color: black;
 background-color: crimson;
}
```

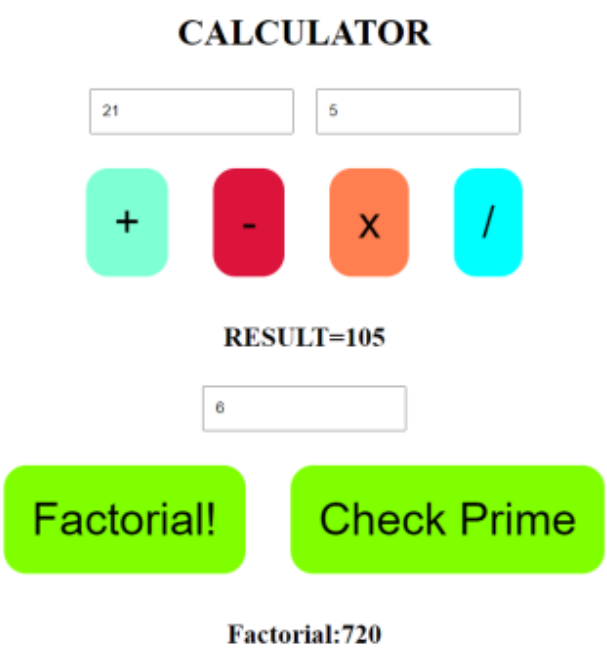
20. subtraction.component.html

```html
<button (click)="subtract(num1,num2)">-</button>
```

21. subtraction.component.ts

```ts
import { Component, EventEmitter, Input, Output } from '@angular/core';

@Component({
 selector: 'app-subtraction',
 templateUrl: './subtraction.component.html',
 styleUrls: ['./subtraction.component.css']
})

export class SubtractionComponent{

 @Input() num1!:string;
 @Input() num2!:string;

 res!:number;

 @Output() sendSubtraction = new EventEmitter<number>();

 subtract(first: string, second: string)
 {
 this.res= parseInt(first) - parseInt(second);
 this.sendSubtraction.emit(this.res);
 }
}
```

OUTPUT :



NAME : SAMARTH AGARWALLA
REGD NO: 201900058
SEC : A