Calculator app using Angular

For

Scripting Language Lab (CS1567)

Name: Reekraj Roy

Registration No: 201900316

Section: B

Contents

Description	**Pg. No.**
Source Code of Components	3-10
Screenshot of Application	10-14

Source Code of Components

Default Components

app-routing.module.ts

```
import { NgModule } from '@angular/core';
import { RouterModule, Routes } from '@angular/router';

const routes: Routes = [];

@NgModule({
  imports: [RouterModule.forRoot(routes)],
  exports: [RouterModule]
})
export class AppRoutingModule { }
```

app.component.html

```
<h1>
  Welcome to {{title}}!
</h1>
<calculator></calculator>
```

app.component.spec.ts

```
import { TestBed } from '@angular/core/testing';
import { RouterTestingModule } from '@angular/router/testing';
import { AppComponent } from './app.component';

describe('AppComponent', () => {
  beforeEach(async () => {
```

```
    await TestBed.configureTestingModule({
      imports: [
        RouterTestingModule
      ],
      declarations: [
        AppComponent
      ],
    }).compileComponents();
  });

  it('should create the app', () => {
    const fixture = TestBed.createComponent(AppComponent);
    const app = fixture.componentInstance;
    expect(app).toBeTruthy();
  });

  it(`should have as title 'calculator'`, () => {
    const fixture = TestBed.createComponent(AppComponent);
    const app = fixture.componentInstance;
    expect(app.title).toEqual('calculator');
  });

  it('should render title', () => {
    const fixture = TestBed.createComponent(AppComponent);
    fixture.detectChanges();
    const compiled = fixture.nativeElement as HTMLElement;
    expect(compiled.querySelector('.content span')?.textContent).toContain('calcu
lator app is running!');
  });
});
```

app.component.ts

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

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

app.module.ts

```
import { NgModule } from '@angular/core';
import { BrowserModule } from '@angular/platform-browser';

import { AppRoutingModule } from './app-routing.module';
import { AppComponent } from './app.component';
import { CalculatorComponent } from './calculator/calculator.component';

@NgModule({
  declarations: [
    AppComponent,
    CalculatorComponent
  ],
  imports: [
    BrowserModule,
    AppRoutingModule
  ],
  providers: [],
  bootstrap: [AppComponent]
})
export class AppModule { }
```

Calculator Component

calculator.component.css

```
@import url(https://fonts.googleapis.com/css?family=Pacifico|Open+Sans|Montserrat
);

h1 {
  margin-top: 0px;
  font-size: 2em;
  font-weight: bold;
}
```

```css
.calculator {
  background-color: rgb(243, 238, 231);
  position: relative;
  margin: 7% auto auto auto;
  width: 400px;
  text-align: center;
  border-color: rgb(99, 94, 97);
  border-radius: 20px;
  box-shadow: inset 2px 0px 10px -5px rgb(85, 80, 83),
    inset -5px 0px 15px -5px rgb(85, 80, 83),
    inset 10px 0px 10px -12px rgb(85, 80, 83),
    inset -2px 0px 4px 2px rgb(85, 80, 83),
    inset -2px 0px 10px 0px rgb(85, 80, 83);
}
I .result {
  border-radius: 8px;
}

button {
  border-radius: 8px;
  width: 80px;
  height: 65px;
  font-size: 20px;
  border: 2px solid;
  border-color: rgb(87, 85, 85);
  margin: 2px 2px 8px 2px;
  transition-duration: 0.45;
  box-shadow: inset 5px 0 15px 5px rgb(109, 103, 106),
    inset -3px 15px 5px rgb(109, 103, 106),
    inset 0px -5px 0px -3px rgb(109, 103, 106),
    inset 0 2px 2px rgb(109, 103, 106), inset 0 5px 15px rgb(109, 103, 106);
}

.long {
  width: 170px;
}

.responsive:hover {
  background-color: rgb(138, 134, 134);
  color: rgb(247, 240, 240);
  box-shadow: inset 5px 0 15px -5px rgb(199, 192, 182),
    inset -3px 0 15px 5px rgb(199, 192, 182),
    inset 0px -5px 0px -3px rgb(199, 192, 182),
    inset 0 0 2px 2px rgb(199, 192, 182), inset 0 5px 15px rgb(199, 192, 182);
```

```
}

input {
  width: 350px;
  height: 60px;
  text-align: right;
  font-size: 1.7em;
}
```

calculator.component.html

```
<div class="calculator">
    <h1>Calculator</h1>
    <div>
        <input class="result" type="text" readonly value="{{ result }}">
    </div>
    <div>
        <button class="responsive long" *ngFor="let longButton of longButtons" (c
lick)="addToExpression(longButton)">
            {{ longButton }}
        </button>
        <button class="responsive" *ngFor="let button of buttons" (click)="addToE
xpression(button)">
            {{ button }}
        </button>
    </div>
</div>
```

calculator.component.spec.ts

```
import { ComponentFixture, TestBed } from '@angular/core/testing';

import { CalculatorComponent } from './calculator.component';

describe('CalculatorComponent', () => {
  let component: CalculatorComponent;
  let fixture: ComponentFixture<CalculatorComponent>;
```

```
  beforeEach(async () => {
    await TestBed.configureTestingModule({
      declarations: [ CalculatorComponent ]
    })
    .compileComponents();
  });

  beforeEach(() => {
    fixture = TestBed.createComponent(CalculatorComponent);
    component = fixture.componentInstance;
    fixture.detectChanges();
  });

  it('should create', () => {
    expect(component).toBeTruthy();
  });
});
```

calculator.component.ts

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

@Component({
  selector: 'calculator',
  templateUrl: './calculator.component.html',
  styleUrls: ['./calculator.component.css']
})
export class CalculatorComponent {

result: string = '';
longButtons: string[] = ['AC', 'CE', 'Prime', '!'];
buttons: string[] = ['7', '8', '9', '/', '4', '5', '6', '*', '1', '2', '3', '-
', '.', '0', '=', '+'];

private prevValue: string = '';
private curValue: string = '';

addToExpression(value: string){

  if (this.result != ''){
    this.prevValue = this.curValue;
```

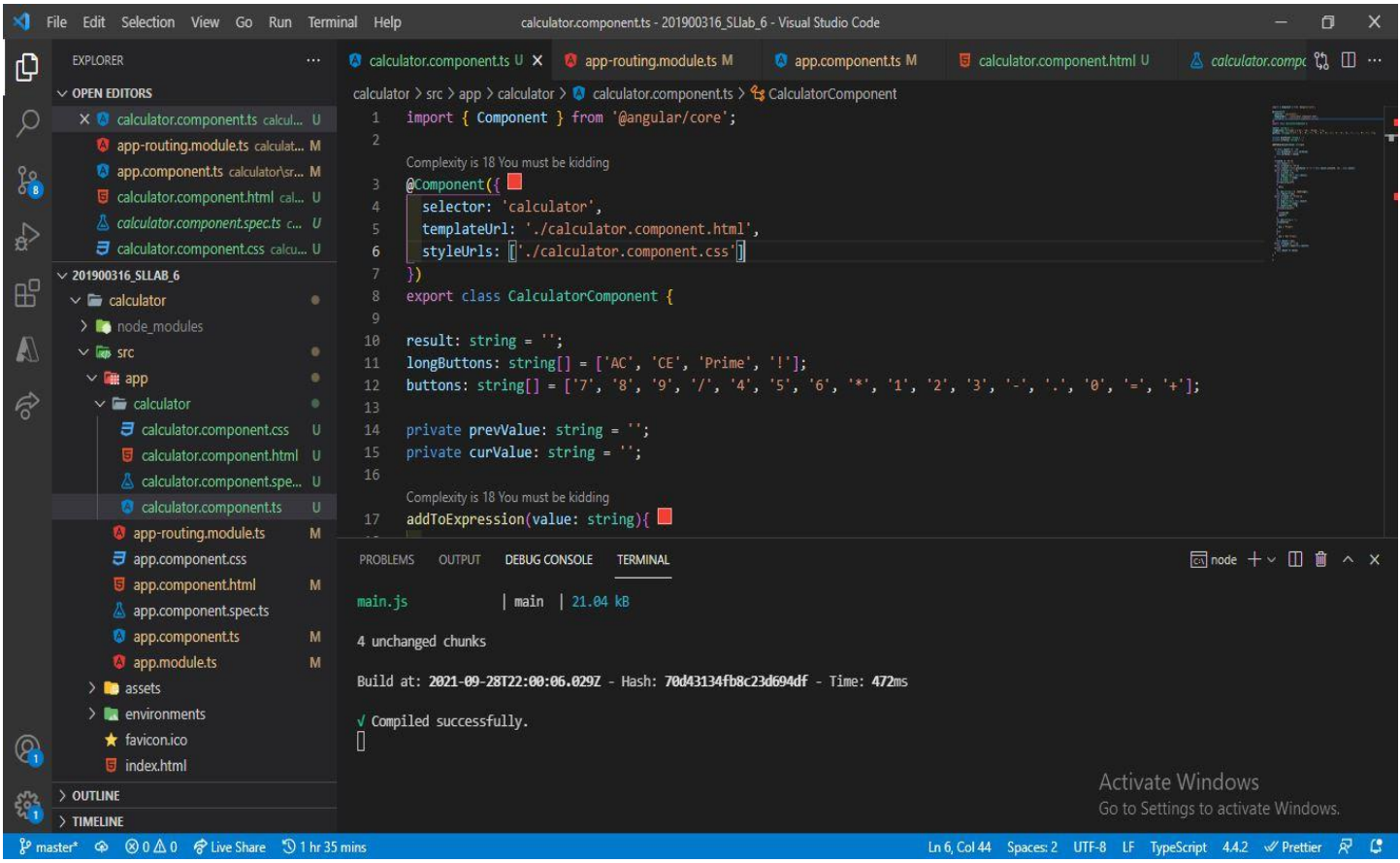
```
    this.curValue = value;
  }

  if(value == 'AC'){
    this.result = '';
  }else if(value == 'CE'){
    this.result = this.prevValue != "=" ? this.result.slice(0, -1) : this.result;
  }else if(value == '!'){
    var i:number = 0;
    var copy:string = this.result;
    var j:number = +copy;
    var k:number = 1;
    for(i=1;i<=j;i++)
    {
      k*=i;
    }
    var ans:string = k .toString();
    this.result = ans;
  }else if(value == 'Prime'){
    var i:number = 0;
    var copy:string = this.result;
    var j:number = +copy;
    var count:number = 0;
    for(i=2;i<=j;j++)
    {
      if(j%i==0)
      count++;
    }
    var ans:string = '';
    if(count==2)
    {
      ans = "Prime";
    }
    else
    {
      ans = "Not Prime";
    }
    this.result = ans;
  }else if(value == '='){
    this.result = eval(this.result);
  }else{
    this.result += value;
  }
}
```

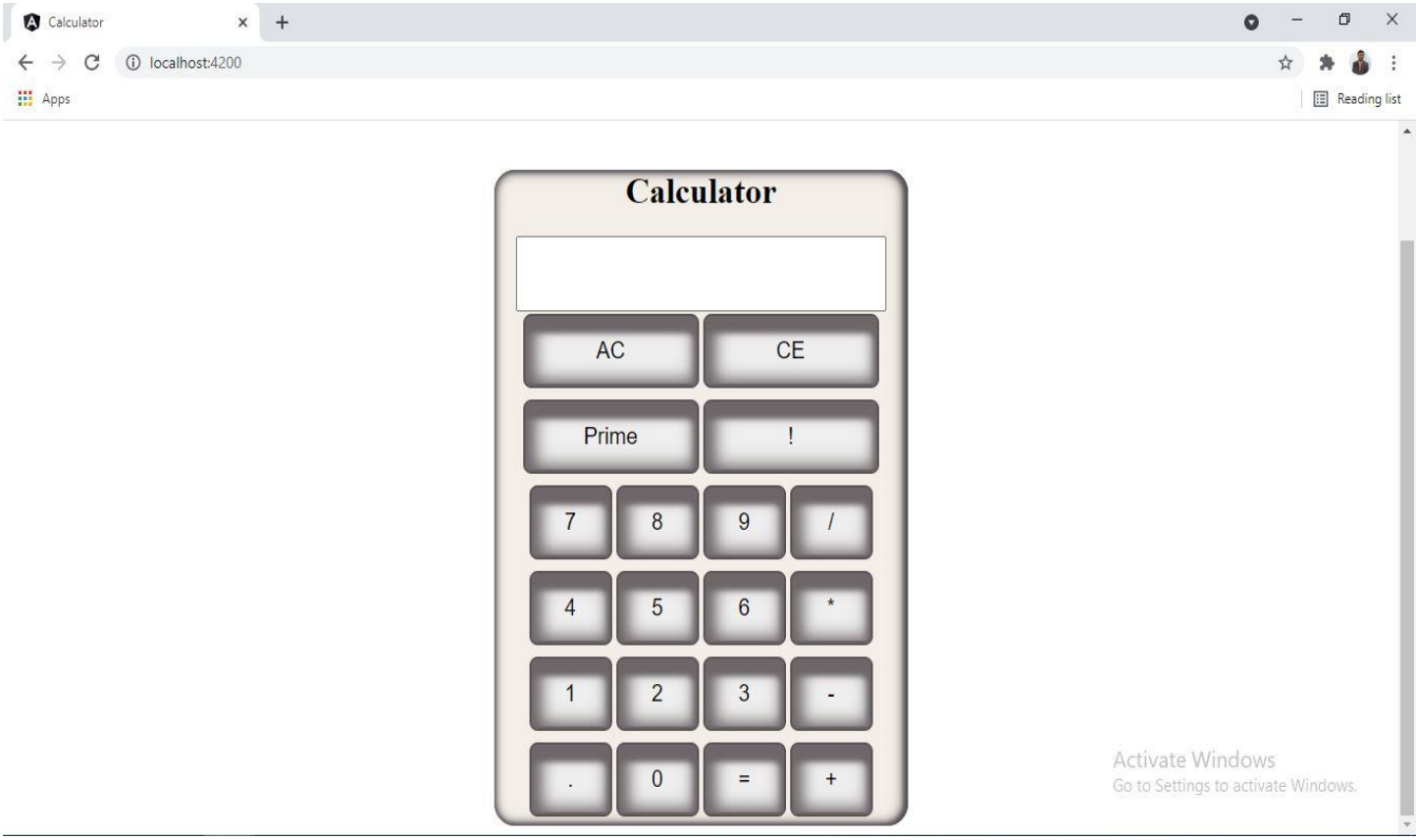
```
}
```

Screenshots

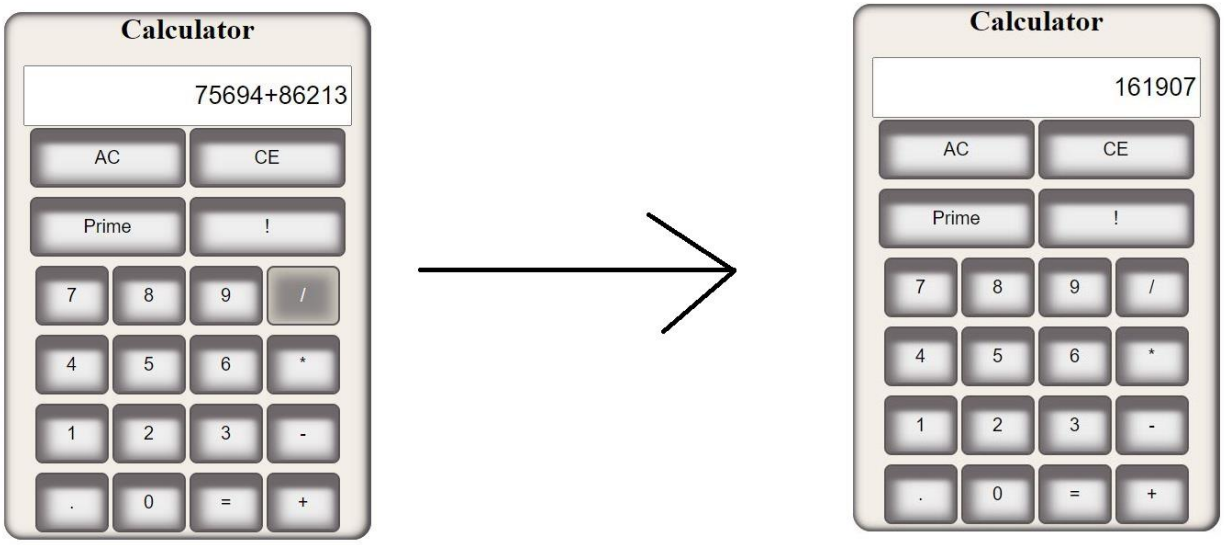
Code Editor



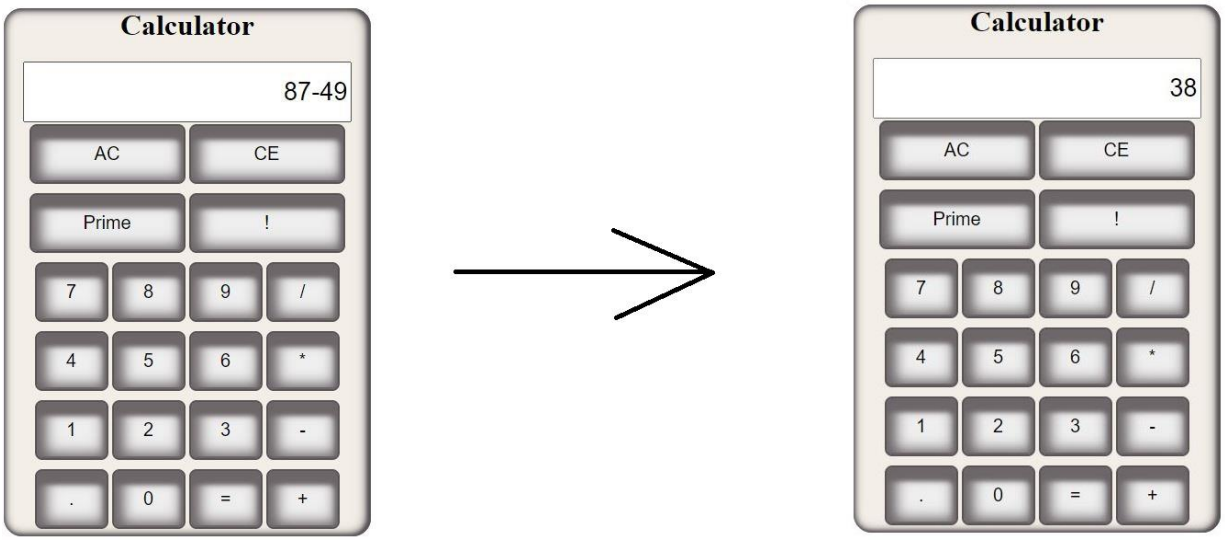
Calculator Application



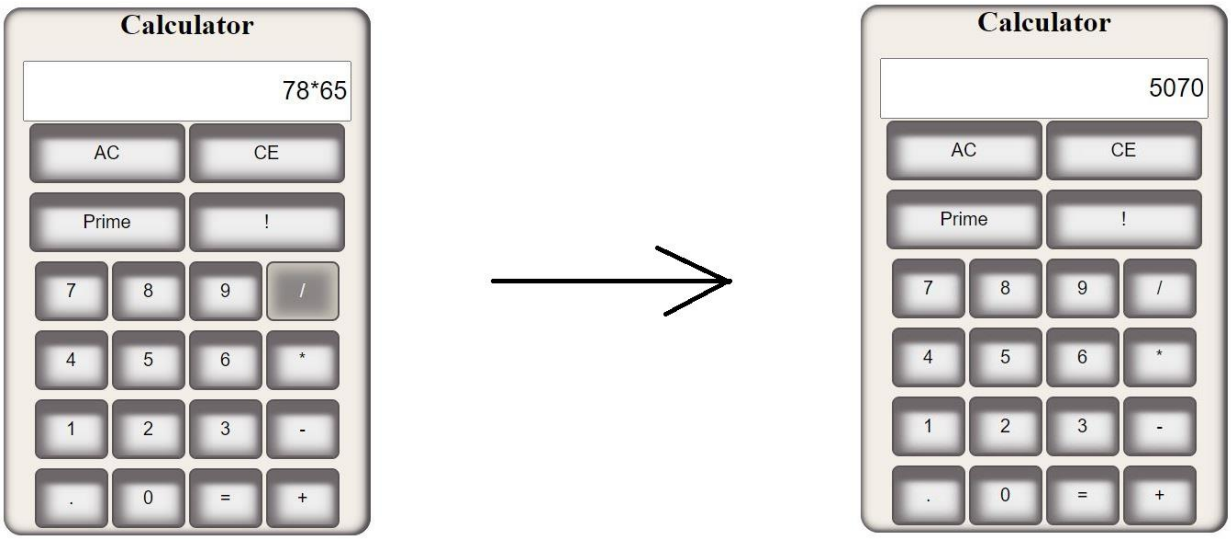
Addition



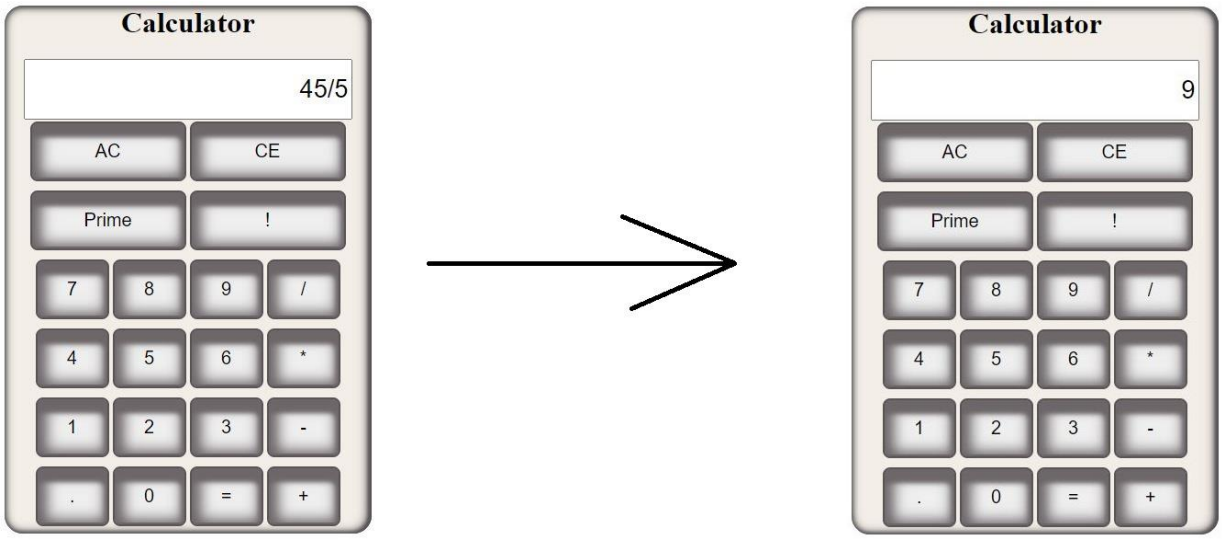
Subtraction



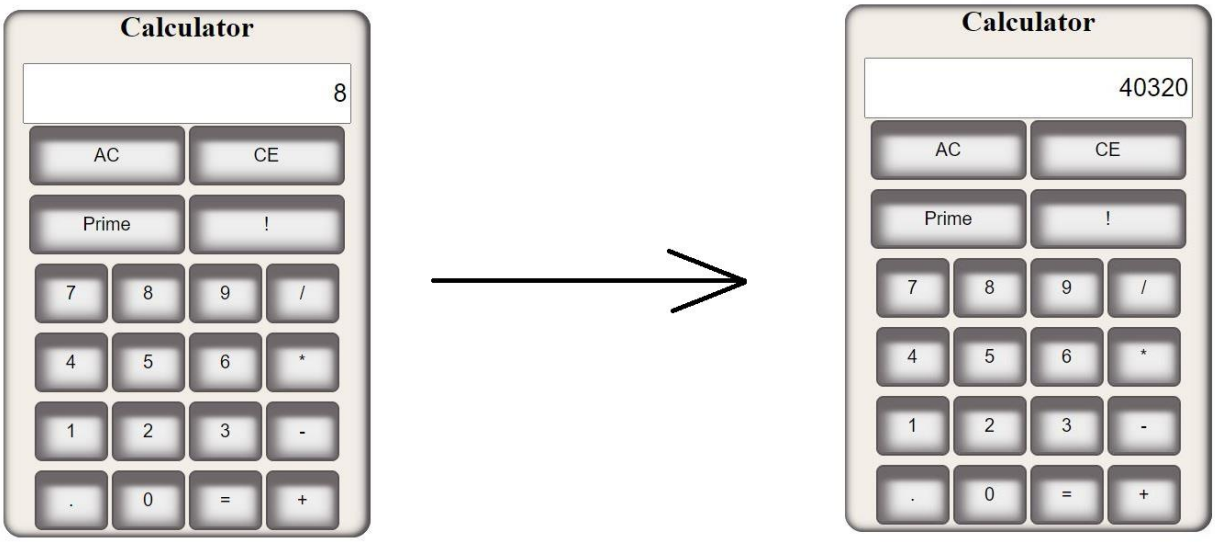
Multiplication



Division



Factorial



Prime Number