The Modular GCD Algorithm (main idea)

```
> g := x^2-7*x+15;
```

$$g := x^2 - 7x + 15 \quad (1)$$

```
> A := expand( g * (x^2+18*x+5) );
```

$$A := x^4 + 11x^3 - 106x^2 + 235x + 75 \quad (2)$$

```
> B := expand( g * (x^2+x+5) );
```

$$B := x^4 - 6x^3 + 13x^2 - 20x + 75 \quad (3)$$

```
> p1 := 11;
```

$$p1 := 11 \quad (4)$$

```
> g1 := Gcd( A mod p1, B mod p1 ) mod p1;
```

$$g1 := x^2 + 4x + 4 \quad (5)$$

```
> p2 := 13;
```

$$p2 := 13 \quad (6)$$

```
> g2 := Gcd( A mod p2, B mod p2 ) mod p2;
```

$$g2 := x^2 + 6x + 2 \quad (7)$$

```
> G := chrem( [g1,g2], [p1,p2] );
```

$$G := x^2 + 136x + 15 \quad (8)$$

Put the coefficient s of G in the symmetric range for the integers modulo M

```
> M := p1*p2;
```

$$M := 143 \quad (9)$$

```
> G := mods(G,M);
```

$$G := x^2 - 7x + 15 \quad (10)$$

```
> gcd(A,B);
```

$$x^2 - 7x + 15 \quad (11)$$