> Question 2

```
> a_1 := 58*x^4-415*x^3-111*x+213;
  b_1 := 69*x^3-112*x^2+413*x+113;
```

$$a\_1 := 58\,x^4 - 415\,x^3 - 111\,x + 213$$

$$b\_1 := 69\,x^3 - 112\,x^2 + 413\,x + 113 \quad (1)$$

```
> a_2 := x^5-111*x^4+ 112*x^3+ 8*x^2-888*x+ 896;
  b_2 := x^5-114*x^4+ 448*x^3-672*x^2+ 669*x-336;
```

$$a\_2 := x^5 - 111\,x^4 + 112\,x^3 + 8\,x^2 - 888\,x + 896$$

$$b\_2 := x^5 - 114\,x^4 + 448\,x^3 - 672\,x^2 + 669\,x - 336 \quad (2)$$

```
> a_3 := 396*x^5-36*x^4+ 3498*x^3-2532*x^2+ 2844*x-1870;
  b_3 := 156*x^5+ 69*x^4+ 1371*x^3-332*x^2+ 593*x-697;
```

$$a\_3 := 396\,x^5 - 36\,x^4 + 3498\,x^3 - 2532\,x^2 + 2844\,x - 1870$$

$$b\_3 := 156\,x^5 + 69\,x^4 + 1371\,x^3 - 332\,x^2 + 593\,x - 697 \quad (3)$$

```
> MignotteBound := proc(f,x)
      local d;
      d := degree(f,x);
      return 2^d*ceil(sqrt(d+1))*maxnorm(f);
  end;
```

$$MignotteBound := \mathbf{proc}(f, x)$$
$$\quad \mathbf{local}\ d;$$
$$\quad d := degree(f, x);\ \mathbf{return}\ 2\hat{}d * \mathrm{ceil}(\mathrm{sqrt}(d+1)) * maxnorm(f)$$
$$\mathbf{end\ proc} \quad (4)$$

```
> my_modular_gcd := proc(a, b)
      local bound, gm, p, M, gcd_p, G, u, g;
      gm := igcd(lcoeff(a), lcoeff(b));
      bound := 2*gm*min(MignotteBound(a,x), MignotteBound(b,x));
      bad_primes := [];
      unlucky_primes := [];
      selected_primes := [];
      p := 19;
      M := 1;
      G := 0;
      while (M <= bound) do
          p := nextprime(p);
          if irem(lcoeff(a), p) = 0 then
              bad_primes := [op(bad_primes), p];
          else
              gcd_p := Gcd(a, b) mod p;
              if gcd_p = 1 then return 1; fi;

              gcd_p := (gm mod p)* gcd_p mod p;

              if G = 0 then
                  G := gcd_p;
                  M := p;
                  selected_primes := [op(selected_primes), p];
              elif degree(gcd_p) > degree(G) then
                  unlucky_primes := [op(unlucky_primes), p];
              elif degree(gcd_p) < degree(G) then
```

```
                unlucky_primes := [op(unlucky_primes), op
  (selected_primes)];
                selected_primes := [];
                G := gcd_p;
                M := p;
            else
                selected_primes := [op(selected_primes), p];
                u := mods(chrem([G, gcd_p], [M, p]), M*p);
                if u = G then
                    g := u/content(u);
                    if divide(a, g) and divide(b, g) then
                        return (g, bad_primes, unlucky_primes,
  selected_primes);
                    fi;
                fi;
                G := u; M := M*p;

            fi;
        fi;
    od;
    return false;
  end;
```

Warning, `bad_primes` is implicitly declared local to procedure `my_modular_gcd`

Warning, `unlucky_primes` is implicitly declared local to procedure `my_modular_gcd`

Warning, `selected_primes` is implicitly declared local to procedure `my_modular_gcd`

$$my\_modular\_gcd := \mathbf{proc}(a, b) \tag{5}$$

local *bound*, *gm*, *p*, *M*, *gcd_p*, *G*, *u*, *g*, *bad_primes*, *unlucky_primes*, *selected_primes*;

gm := *igcd*(*lcoeff*(*a*), *lcoeff*(*b*));

bound := 2 * *gm* * min(*MignotteBound*(*a*, *x*), *MignotteBound*(*b*, *x*));

bad_primes := [ ];

unlucky_primes := [ ];

selected_primes := [ ];

p := 19;

M := 1;

G := 0;

while *M* <= *bound* **do**

p := *nextprime*(*p*);

if *irem*(*lcoeff*(*a*), *p*) = 0 **then**

bad_primes := [*op*(*bad_primes*), *p*]

else

gcd_p := *Gcd*(*a*, *b*) **mod** *p*;

if *gcd_p* = 1 **then return** 1 **end if**;

gcd_p := (*gm* **mod** *p*) * *gcd_p* **mod** *p*;

if *G* = 0 **then**

G := *gcd_p*; *M* := *p*; *selected_primes* := [*op*(*selected_primes*), *p*]

elif *degree*(*G*) < *degree*(*gcd_p*) **then**

```
            unlucky_primes := [op(unlucky_primes), p]
        elif degree(gcd_p) < degree(G) then
            unlucky_primes := [op(unlucky_primes), op(selected_primes)];
            selected_primes := [ ];
            G := gcd_p;
            M := p
        else
            selected_primes := [op(selected_primes), p];
            u := mods(chrem([G, gcd_p], [M, p]), M*p);
            if u = G then
                g := u / content(u);
                if divide(a, g) and divide(b, g) then
                    return g, bad_primes, unlucky_primes, selected_primes
                end if
            end if;
            G := u;
            M := M*p
        end if
    end if
  end do;
  return false
end proc
```

```
> my_modular_gcd(a_1, b_1);
```

$$1 \tag{6}$$

```
> my_modular_gcd(a_2, b_2);
```

$$x^2 - 111\,x + 112, [\,], [29, 31], [23, 37, 41] \tag{7}$$

```
> my_modular_gcd(a_3, b_3);
```

$$3\,x^3 + 24\,x - 17, [\,], [\,], [23, 29, 31] \tag{8}$$

```
> gcd(a_1, b_1);
```

$$1 \tag{9}$$

```
> gcd(a_2, b_2);
```

$$x^2 - 111\,x + 112 \tag{10}$$

```
> gcd(a_3, b_3);
```

$$1 \tag{11}$$

```
> gcd_p := Gcd(a_2, b_2) mod 29;
```

$$gcd\_p := x^3 + 7\,x^2 + 6\,x + 21 \tag{12}$$