```
> restart:
> mig_Bound := proc(param_f::anything, param_x::anything)

               local deg_var:
               deg_var := degree(param_f, param_x):
               return 2^deg_var*ceil(sqrt(deg_var+1))*maxnorm
  (param_f):

               end proc:
> prim_check := proc(param_a::polynom, param_b::polynom)

                local A, B:
                A, B := param_a, param_b:
                if content(A,x) = 1 and content(B,x) = 1 then
                    return 'PASS':
                else:
                    return 'FAIL':
                fi:

                end proc:
> printf("\nInput Polynomials:\n"):
  a_One := 58*x^4 - 415*x^3 - 111*x + 213;
  b_One := 69*x^3 - 112*x^2 + 413*x + 113;
```

Input Polynomials:

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

$$b\_One := 69\,x^3 - 112\,x^2 + 413\,x + 113 \tag{1}$$

```
> content(a_One,x);
  content(b_One,x);
  printf("\nHence primitive.\n"):
```

$$1$$

$$1$$

Hence primitive.

```
> bound_One:= min(mig_Bound(a_One, x), mig_Bound(b_One, x)):
  printf("\nThe value of B is: %d.\n", bound_One):
```

The value of B is: 6608.

```
> p_List_One := [23, 29, 31]:

  M := 1:
```

```
for i from 1 to nops(p_List_One) do
    M := M*p_List_One[i]:
od:
printf("\nThe value of M (product of the primes) is: %d.\n", M):
```

The value of M (product of the primes) is: 20677.

```
> beta_one := igcd(lcoeff(a_One), lcoeff(b_One));
  printf("For this case there are no bad primes.\n"):
```

$$beta\_one := 1$$

For this case there are no bad primes.

```
>
  printf("\nInput Polynomials:\n"):
  a_Two := x^5 - 111*x^4 + 112*x^3 + 8*x^2 - 888*x + 896;
  b_Two := x^5 - 114*x^4 + 448*x^3 - 672*x^2 + 669*x - 336;
```

Input Polynomials:

$$a\_Two := x^5 - 111\,x^4 + 112\,x^3 + 8\,x^2 - 888\,x + 896$$
$$b\_Two := x^5 - 114\,x^4 + 448\,x^3 - 672\,x^2 + 669\,x - 336 \quad \textbf{(2)}$$

```
> content(a_Two, x);
  content(b_Two, x);
  printf("\nHence primitive.\n"):
```

$$1$$
$$1$$

Hence primitive.

```
> bound_Two := min(mig_Bound(a_Two, x), mig_Bound(b_Two, x)):
  printf("\nThe value of B is: %d.\n", bound_Two):
```

The value of B is: 64512.

```
> beta_Two := igcd(lcoeff(a_Two), lcoeff(b_Two));
  printf("For this case there are no bad primes.\n"):
```

$$beta\_Two := 1$$

For this case there are no bad primes.

```
> p_List_Two := [23, 29, 31, 37, 41, 43];
  g_One := Gcd(a_Two, b_Two) mod p_List_Two[1];
  g_Two := Gcd(a_Two, b_Two) mod p_List_Two[2];
  g_Three := Gcd(a_Two, b_Two) mod p_List_Two[3];
  g_Four := Gcd(a_Two, b_Two) mod p_List_Two[4];
  g_Five := Gcd(a_Two, b_Two) mod p_List_Two[5];
  g_Six := Gcd(a_Two, b_Two) mod p_List_Two[6];
```

```
printf("\nFor this case, p = 29 and p = 31 are bad primes.\n"):
```

$$p\_List\_Two := [23, 29, 31, 37, 41, 43]$$
$$g\_One := x^2 + 4x + 20$$
$$g\_Two := x^3 + 7x^2 + 6x + 21$$
$$g\_Three := x^3 + 23x^2 + 25x + 4$$
$$g\_Four := x^2 + 1$$
$$g\_Five := x^2 + 12x + 30$$
$$g\_Six := x^2 + 18x + 26$$

```
For this case, p = 29 and p = 31 are bad primes.
```

```
> mod_P_List_One := [23, 37, 41, 43]:
  mod_M := 1:
  for i from 1 to nops(mod_P_List_One) do
      mod_M := mod_M*mod_P_List_One[i]:
  od:
  printf("\nThe value of M (product of the primes) is: %d.\n", mod_M)
  :
```

```
The value of M (product of the primes) is: 1500313.
```

```
> g_comp := mods(chrem([g_One, g_Four, g_Five, g_Six], [23, 37, 41,
  43]), mod_M);
  g_comp_prim := primpart(g_comp);
  actual_gcd := gcd(a_Two, b_Two);
```

$$g\_comp := x^2 - 111x + 112$$
$$g\_comp\_prim := x^2 - 111x + 112$$
$$actual\_gcd := x^2 - 111x + 112 \quad (3)$$

```
> divide(a_Two, g_comp_prim), divide(b_Two, g_comp_prim);
```

$$true, true \quad (4)$$

```
> printf("\nInput Polynomials:\n"):
  a_Three := 396*x^5 - 36*x^4 + 3498*x^3 - 2532*x^2 + 2844*x - 1870;
  b_Three := 156*x^5 + 69*x^4 + 1371*x^3 - 332*x^2 + 593*x - 697;
```

```
Input Polynomials:
```

$$a\_Three := 396x^5 - 36x^4 + 3498x^3 - 2532x^2 + 2844x - 1870$$
$$b\_Three := 156x^5 + 69x^4 + 1371x^3 - 332x^2 + 593x - 697 \quad (5)$$

```
> content(a_Three,x);
  content(b_Three,x);
  printf("\nHence not primitive.\n"):
```

$$2$$

$$1$$

Hence not primitive.

```
> new_A_Three := primpart(a_Three):
  content(new_A_Three, x);
  content(b_Three, x);
  new_A_Three;
```

$$1$$

$$1$$

$$198\,x^5 - 18\,x^4 + 1749\,x^3 - 1266\,x^2 + 1422\,x - 935 \tag{6}$$

```
> beta_Three := igcd(lcoeff(new_A_Three), lcoeff(b_Three));
  printf("For this case there are no bad primes.\n"):
```

$$beta\_Three := 6$$

For this case there are no bad primes.

```
> p_List_Three := [23, 29, 31, 37, 41, 43];
  g_m_1 := Gcd(new_A_Three, b_Three) mod p_List_Three[1]:
  g_m_1 := beta_Three * g_m_1 mod p_List_Three[1];
  g_m_2 := Gcd(new_A_Three, b_Three) mod p_List_Three[2]:
  g_m_2 := beta_Three * g_m_2 mod p_List_Three[2];
  g_m_3 := Gcd(new_A_Three, b_Three) mod p_List_Three[3]:
  g_m_3 := beta_Three * g_m_3 mod p_List_Three[3];
  g_m_4 := Gcd(new_A_Three, b_Three) mod p_List_Three[4]:
  g_m_4 := beta_Three * g_m_4 mod p_List_Three[4];
  g_m_5 := Gcd(new_A_Three, b_Three) mod p_List_Three[5]:
  g_m_5 := beta_Three * g_m_5 mod p_List_Three[5];
  g_m_6 := Gcd(new_A_Three, b_Three) mod p_List_Three[6]:
  g_m_6 := beta_Three * g_m_6 mod p_List_Three[6];
```

$$p\_List\_Three := [23, 29, 31, 37, 41, 43]$$

$$g\_m\_1 := 6\,x^3 + 2\,x + 12$$

$$g\_m\_2 := 6\,x^3 + 19\,x + 24$$

$$g\_m\_3 := 6\,x^3 + 17\,x + 28$$

$$g\_m\_4 := 6\,x^3 + 11\,x + 3$$

$$g\_m\_5 := 6\,x^3 + 7\,x + 7$$

$$g\_m\_6 := 6\,x^3 + 5\,x + 9 \tag{7}$$

```
> mod_P_List_Three := [23, 29, 31, 37, 41, 43]:
  mod_M_Two := 1:
  for i from 1 to nops(mod_P_List_One) do
      mod_M_Two := mod_M_Two*mod_P_List_Three[i]:
  od:
  printf("\nThe value of M (product of the primes) is: %d.\n",
```

```
mod_M_Two):
```

The value of M (product of the primes) is: 765049.

```
> bound_Three := min(mig_Bound(new_A_Three, x), mig_Bound(b_Three, x)
  ):
  printf("\nThe value of B is: %d.\n", bound_Three):
```

The value of B is: 131616.

```
> g_Comp_Three := mods(chrem([g_m_1, g_m_2, g_m_3, g_m_4, g_m_5,
  g_m_6], [23, 29, 31, 37, 41, 43]), mod_M_Two);
  g_Comp_Prim_Three := primpart(g_Comp_Three);
  actual_Gcd_Three := gcd(new_A_Three,b_Three);
```

$$g\_Comp\_Three := 6\,x^3 + 48\,x - 34$$

$$g\_Comp\_Prim\_Three := 3\,x^3 + 24\,x - 17$$

$$actual\_Gcd\_Three := 3\,x^3 + 24\,x - 17 \tag{8}$$

```
> divide(new_A_Three, g_Comp_Prim_Three), divide(b_Three,
  g_Comp_Prim_Three);
```

$$true, true \tag{9}$$