```
> restart:
```

# Question 3a:

```
> poly_A := expand((x^4-3*x^3*y-x^2-y)*(8*x-4*y+12)*(2*y^2-2)):
> (* Proc for calculating GCD *)

  rec_gcd := proc(a::anything)
      local l, g, i:
      l := a:
      g := l[1]:

      for i from 2 to nops(l) - 1 do
         g := gcd(g, l[i]):
      od:

    printf("\nThe computed GCD of coeffs is: %a.\n\n", g);
  end proc:
> pc_x := content(poly_A,x,'pp'):
  pp_x := pp:

  (* GCD of coeffs in Z[y][x] *)

  co_x := coeffs(poly_A,x):
  co_x_list := [co_x]:
  rec_gcd(co_x_list);

  printf("Considering x as the main variable:\nThe content of the
  polynomial is: %a.\n\nThe primitive part of the polynomial
  is:\n%a.\n", pc_x,pp_x);

  pc_y := content(poly_A,y,'pp'):
  pp_y := pp:

  (* GCD of coeffs in Z[x][y] *)

  co_x_2 := coeffs(poly_A,y):
  co_x_2_list := [co_x_2]:
  rec_gcd(co_x_2_list);

  printf("Considering y as the main variable:\nThe content of the
  polynomial is: %a.\n\nThe primitive part of the polynomial
  is:\n%a.\n", pc_y,pp_y);
```

```
The computed GCD of coeffs is: 8*y^2-8.
```

```
Considering x as the main variable:
The content of the polynomial is: 8*y^2-8.

The primitive part of the polynomial is:
2*x^5-7*x^4*y+3*x^3*y^2+3*x^4-9*x^3*y-2*x^3+x^2*y-3*x^2-2*x*y+y^2-3*
y.

The computed GCD of coeffs is: 8.

Considering y as the main variable:
The content of the polynomial is: 8.

The primitive part of the polynomial is:
2*x^5*y^2-7*x^4*y^3+3*x^3*y^4+3*x^4*y^2-9*x^3*y^3-2*x^5+7*x^4*y-5*
x^3*y^2+x^2*y^3-3*x^4+9*x^3*y-3*x^2*y^2-2*x*y^3+y^4+2*x^3-x^2*y-3*
y^3+3*x^2+2*x*y-y^2+3*y.
```

## Question 3b:

```
> poly_a := expand(3*x^3+(y+1)*x):
  poly_b := expand((2*y)*x^2+2*x+y):

  p_rem := prem(poly_a, poly_b, x, 'm','q'):
  printf("\nThe multiplier is: %a.\nThe pseudo-remainder is: %a.
  \nThe pseudo-quotient is: %a.\n\n",m, p_rem, q);
```

```
The multiplier is: 4*y^2.
The pseudo-remainder is: 4*x*y^3-2*x*y^2+12*x+6*y.
The pseudo-quotient is: 6*x*y-6.
```

## Question 3c:

```
> (* Primitive Euclidiean Algorithm Implementation *)

  prim_algo := proc(a::polynom, b::polynom)

  local A, B, k, r, g, pr:
  A, B, k := a, b, 1:
  r[0] := primpart(A,x):
  r[1] := primpart(B,x):

  while r[k] <> 0 do
     pr := prem(r[k-1],r[k],x):
     r[k+1] := primpart(pr,x):
     k := k+1;
  od:

  g := r[k-1];
  g := -1*expand((gcd(content(a,x),content(b,x))*g));
  return printf("The GCD using prim E algo is: %a.\n", g);
  end proc:
```

```
> a := expand((x^4-3*x^3*y-x^2-y)*(2*x-y+3)*(8*y^2-8)):
  b := expand((x^3*y^2+x^3+x^2+3*x+y)*(2*x-y+3)*(12*y^3-12)):
  prim_algo(a,b);

The GCD using prim E algo is: 8*x*y-4*y^2-8*x+16*y-12.
```

```
> gcd(a,b);
```

$$8xy-4y^2-8x+16y-12 \tag{1}$$