```
> restart:
> (* HELPER FUNCTION *)

  CHECK := proc(var_a,var_b,var_c,var_d)

      local a,b,c,p:
      a := var_a:
      b := var_b:
      c := var_c:
      p := var_d:

      if a-Expand(b*c) mod p = 0 then
          return "PASS":
      else
          return "FAIL":
      fi:

  end proc:

> (* MAIN FUNCTION *)

  ALGORITHM := proc(var_a,var_b,var_c)

      local a,b,p,deg_a,deg_b,total_deg,int_list,eval_list,i,j,
  interp_val,a_j,b_j,c_i,res:
      local res_two,updated_list:
      a := var_a:
      b := var_b:
      p := var_c:
      deg_a := degree(a,y):
      deg_b := degree(b,y):
      total_deg := deg_a-deg_b:
      int_list := [seq(i,i=1..total_deg+1)]:
      eval_list := []:
      updated_list := []:

      for j from 1 to nops(int_list) do
          a_j := Eval(a,y=int_list[j]) mod p:
          b_j := Eval(b,y=int_list[j]) mod p:
          c_i := Quo(a_j,b_j,x) mod p:
          eval_list := [op(eval_list),c_i]:
      od:

      res := Interp(int_list,eval_list,y) mod p:
      res_two := CHECK(a,b,res,p):

      if res_two="PASS" then
          return res:
```

```
    else
        return "FAIL":
    fi:

end proc:
```

```
> (* INPUT ONE *)
  p1 := 101:
  b1 := y*x^2+2*y*x+y^2+3*y:
  q1 := 2*x^2+(3*y^2+2)*x+99*y+1:
  a1 := Expand(b1*q1) mod p1:
```

```
> (* OUTPUT ONE *)
  ALGORITHM(a1,b1,p1);
  Divide(a1,b1,'s1') mod p1;
  s1;
```

$$3\,x\,y^2 + 2\,x^2 + 2\,x + 99\,y + 1$$

$$true$$

$$3\,x\,y^2 + 2\,x^2 + 2\,x + 99\,y + 1 \qquad \textbf{(1)}$$

```
> (* INPUT TWO *)
  p2 := 101:
  b2 := y*x^2+2*y*x+y^2+3*y:
  q2 := 2*x^2+(3*y^2+2)*x+99*y+1:
  a2 :=  Expand(b2*q2+3*y^2*x) mod p2:
```

```
> (* OUTPUT TWO *)
  ALGORITHM(a2,b2,p2);
  Divide(a2,b2,'s2') mod p2;
  s2;
```

"FAIL"

$$false$$

$$s2 \qquad \textbf{(2)}$$