Question 1 (Part c):

```
> restart:
> `mod` := mods:
> (* Finding u congruent cuberoot(a) mod 5 *)

  pol_a := x^6 - 531*x^5 + 94137*x^4 - 5598333*x^3 + 4706850*x^2 -
  1327500*x + 125000;
  pol_a_mod_5 := pol_a mod 5;
  pol_factor := Factor(pol_a_mod_5) mod 5;
  pol_expand := expand(x*(x-2));
  pol_rev := expand((x^2 - 2*x)^3) mod 5;
```

$$pol\_a := x^6 - 531\,x^5 + 94137\,x^4 - 5598333\,x^3 + 4706850\,x^2 - 1327500\,x + 125000$$

$$pol\_a\_mod\_5 := x^6 - x^5 + 2\,x^4 + 2\,x^3$$

$$pol\_factor := x^3\,(x-2)^3$$

$$pol\_expand := x^2 - 2\,x$$

$$pol\_rev := x^6 - x^5 + 2\,x^4 + 2\,x^3 \qquad (1)$$

```
> check_res := proc(pol_a::polynom, pol_b::polynom)

  local a, b:
  a, b := pol_a, pol_b:

  if expand((b)^3) = a then
      return 'PASS':
  else
      return 'FAIL':
  fi:

  end proc:

> mig_bound := proc(pol::polynom, indeterminate_var)

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

  end proc:

> p_adic_cube_root_algo := proc(pol_a::polynom, u_in::polynom,
  prime_p::prime)

  local a, bound, u0, u_tilde, p, e_k, u_k, k, t, d:
```

```
  a, u0, u_tilde, p, e_k, u_k, k, t, d:= pol_a, u_in, u_in, prime_p,
  0, 0, 1, 0, expand(-3*u_in):
  bound := mig_bound(pol_a, x):

  while true do:

      e_k := (a - expand((u_tilde)^3)):
      if e_k = 0 then
          return u_tilde:
      elif p^k > 2*bound then
          return 'FAIL':
      fi:
      t := e_k/p^k:
      t := (-t) mod p:
      u_k := Quo(t, -3*u0^2, x, 'r') mod p:
      if Divide(t, d) mod p = false then
          return 'FAIL':
      fi:
      u_tilde := u_tilde + u_k*p^k:
      k := k + 1:
  od:

  end proc:
```

```
> test_a := x^6 - 531*x^5 + 94137*x^4 - 5598333*x^3 + 4706850*x^2 -
  1327500*x + 125000:
  test_u_in := expand(x*(x-2)):
  test_p := 5:

  printf("\nINPUT LIST 1:\n1) POLYNOMIAL: %a.\n2) INITIAL VALUE: %a.
  \n3) PRIME: %a.\n", test_a, test_u_in, test_p):
```

```
INPUT LIST 1:
1) POLYNOMIAL: x^6-531*x^5+94137*x^4-5598333*x^3+4706850*x^2-1327500*
x+125000.
2) INITIAL VALUE: x^2-2*x.
3) PRIME: 5.
```

```
> cube_root := p_adic_cube_root_algo(test_a, test_u_in, test_p):
  cube_root;
```

$$x^2 - 177\,x + 50 \tag{2}$$

```
> check_res(test_a, cube_root);
```

$$PASS \tag{3}$$

```
> (* Finding u congruent cuberoot(a) mod 5 *)

  pol_b := x^6 - 406*x^5 + 94262*x^4 - 5598208*x^3 + 4706975*x^2 -
  1327375*x + 125125;
```

```
pol_b_mod_5 := pol_b mod 5;
pol_factor_b := Factor(pol_b_mod_5) mod 5;
pol_expand_b := expand(x*(x-2));
pol_rev := expand((x^2 - 2*x)^3) mod 5;
```

$$pol\_b := x^6 - 406\,x^5 + 94262\,x^4 - 5598208\,x^3 + 4706975\,x^2 - 1327375\,x + 125125$$

$$pol\_b\_mod\_5 := x^6 - x^5 + 2\,x^4 + 2\,x^3$$

$$pol\_factor\_b := x^3\,(x-2)^3$$

$$pol\_expand\_b := x^2 - 2\,x$$

$$pol\_rev := x^6 - x^5 + 2\,x^4 + 2\,x^3 \qquad \textbf{(4)}$$

```
> test_b := x^6 - 406*x^5 + 94262*x^4 - 5598208*x^3 + 4706975*x^2 -
  1327375*x + 125125:
  test_u_in_b := expand(x*(x-2)):
  test_p_2 := 5:

  printf("\nINPUT LIST 2:\n1) POLYNOMIAL: %a.\n2) INITIAL VALUE: %a.
  \n3) PRIME: %a.\n", test_b, test_u_in_b, test_p_2):

INPUT LIST 2:
1) POLYNOMIAL: x^6-406*x^5+94262*x^4-5598208*x^3+4706975*x^2-1327375*
x+125125.
2) INITIAL VALUE: x^2-2*x.
3) PRIME: 5.
```

```
> cube_root_b := p_adic_cube_root_algo(test_b, test_u_in_b, test_p_2)
  :
  cube_root_b;
```

$$FAIL \qquad \textbf{(5)}$$