```
> DiophantSolve := proc(a,b,c,x,p)
  local g,sigma,tau,q,s,t;
      g := Gcdex(a,b,x,'s','t') mod p;
      if g <> 1 then error "a and b are not relatively prime!" fi;
      sigma := Rem(c*s,b,x,'q') mod p;
      tau := Expand(c*t+q*a) mod p;
      return (sigma, tau);
  end
```

```
DiophantSolve := proc(a, b, c, x, p)                                    (1)
    local g, sigma, tau, q, s, t;
    g := Gcdex(a, b, x, 's', 't') mod p;
    if g <> 1 then error "a and b are not relatively prime!" end if;
    sigma := Rem(c * s, b, x, 'q') mod p;
    tau := Expand(c * t + q * a) mod p;
    return sigma, tau
end proc
```

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

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

```
> UniHenselLifting := proc(input_a, x, input_u0, input_w0, p)
  local alpha, a, u0, w0, u, w, B, e_k, c, k, u_k, w_k, s, t, r, q;
  `mod` := mods;
      alpha := lcoeff(input_a, x);
      a := alpha * input_a;
      B := alpha * MignotteBound(input_a, x);
      u0 := alpha * (input_u0 / lcoeff(input_u0, x)) mod p;
      w0 := alpha * (input_w0 / lcoeff(input_w0, x)) mod p;
      print(a mod p, u0 mod p, w0 mod p);
      print(expand(a-u0*w0) mod p);
      k := 1;
      s, t := DiophantSolve(w, u, 1, x, p);
      u := u0; w := w0;
      while (a - u*w) <> 0 do
          e_k := expand(a - u*w);
          if e_k = 0 then return (primpart(u), primpart(w)); fi;
          if p^k > 2*B then return FAIL; fi;
          c := (e_k / (p^k)) mod p;
          u_k, w_k := DiophantSolve(w0, u0, c, x, p);
          u := u + u_k * (p^k);
          w := w + w_k * (p^k);
          u := expand(alpha * u / lcoeff(u, x)) mod (p^(k+1));
          w := expand(alpha * w / lcoeff(w, x)) mod (p^(k+1));
          k := k + 1;
      od;
  end;
```

```
UniHenselLifting := proc(input_a, x, input_u0, input_w0, p)                                              (3)
    local alpha, a, u0, w0, u, w, B, e_k, c, k, u_k, w_k, s, t, r, q;
    mod := mods;
    alpha := lcoeff(input_a, x);
    a := alpha * input_a;
    B := alpha * MignotteBound(input_a, x);
    u0 := alpha * input_u0/lcoeff(input_u0, x) mod p;
    w0 := alpha * input_w0/lcoeff(input_w0, x) mod p;
    print(a mod p, u0 mod p, w0 mod p);
    print(expand(a − w0 * u0) mod p);
    k := 1;
    s, t := DiophantSolve(w, u, 1, x, p);
    u := u0;
    w := w0;
    while a − u * w <> 0 do
        e_k := expand(a − u * w);
        if e_k = 0 then return primpart(u), primpart(w) end if;
        if 2 * B < p^k then return FAIL end if;
        c := e_k/p^k mod p;
        u_k, w_k := DiophantSolve(w0, u0, c, x, p);
        u := u + u_k * p^k;
        w := w + w_k * p^k;
        u := expand(alpha * u/lcoeff(u, x)) mod p^(k + 1);
        w := expand(alpha * w/lcoeff(w, x)) mod p^(k + 1);
        k := k + 1
    end do
end proc
```

```
> a  := x^4 - 2*x^3-233*x^2 -214*x +85;
  u0  := x^2-3*x-2;
  w0  := x^2+x+3;
  factor(a);
```

$$a := x^4 - 2\,x^3 - 233\,x^2 - 214\,x + 85$$

$$u0 := x^2 - 3\,x - 2$$

$$w0 := x^2 + x + 3$$

$$\left(x^2 + 15\,x + 17\right)\left(x^2 - 17\,x + 5\right) \qquad (4)$$

```
> expand((a - u0*w0)) mod 7;
```

$$0 \qquad (5)$$

```
> UniHenselLifting(a, x, u0, w0, 7);
```

$$0$$

$$x^2 - 17\,x + 5, x^2 + 15\,x + 17 \qquad (6)$$

```
> b := 48*x^4 - 22*x^3 + 47*x^2 + 144;
```

```
  u0 := x^2+4*x+2;
  w0 := x^2+4*x+5;
  factor(b);
```

$$b := 48\,x^4 - 22\,x^3 + 47\,x^2 + 144$$

$$u0 := x^2 + 4\,x + 2$$

$$w0 := x^2 + 4\,x + 5$$

$$\left(6\,x^2 - 11\,x + 12\right)\,\left(8\,x^2 + 11\,x + 12\right) \qquad (7)$$

```
> `mod` := mods;
  6 mod 7;
  48 mod 7;
```

$$mod := mods$$

$$-1$$

$$-1 \qquad (8)$$

```
> UniHenselLifting(b, x, 6*u0, w0, 7);
```

$$x^4 + x^3 + 2\,x^2 + 3,\ -x^2 + 3\,x - 2,\ -x^2 + 3\,x + 2$$

$$0$$

$$6\,x^2 - 11\,x + 12,\ 8\,x^2 + 11\,x + 12 \qquad (9)$$