ALGORITHME D'EUCLIDE ÉTENDU RAPIDE

Pour constuire des vecteurs et de matrices, on pourra utiliser les commandes $\texttt{Vector}(n, [v_1, \ldots, v_n])$ et

$$\texttt{Matrix}(n, m, [[a_{1,1}, \ldots, a_{1,m}], \ldots, [a_{n,1}, \ldots, a_{n,m}]]) \ .$$

Pour multiplier deux matrices A et B on pourra utiliser `A.B`. Pour calculer le degré d'un polynôme P on pourra utiliser `degree(normal(P))`.

Éssayer sur les exemples suivants :

$$\begin{aligned} f &= x^8 + 5x^7 + 3x^6 + 5x^4 + 5x^3 + 5x^2 + 2x + 2 \\ g &= x^7 + 4x^6 + 4x^5 + 2x^4 + x^3 + 5x^2 + x + 3 \end{aligned}$$

On construit la suite de polyômes de $\mathbb{Z}[X]$ définie par :

$$\begin{aligned} P_0 &= 1 \\ P_1 &= X + 1 \\ P_{n+2} &= XP_{n+1} + P_n \end{aligned}$$

Comparer l'algorithme rapide et l'algorithme normal pour P_{n+1}, P_n.

Écrire une procédure qui implante l'algorithme du pgcd étendu rapide pour deux polynômes.

```
tr:=proc(f::polynom,k)
  local d;
  d:=degree(normal(f));
  if d<k then return f*x^(k-d);
  else return quo(f,x^(d-k),x);
  end if;
end proc;

pgcd:= proc(f::polynom,g::polynom,k)
  local df,dg,dr,q::polynom,r::polynom,rho,R::vector(polynom),
        M::matrix(polynom),N::matrix(polynom),d,dd;
  df:=degree(normal(f)); dg:=degree(normal(g));
  if g=0 or k < df-dg then
    return Matrix(2,2,[[1,0],[0,1]]);
  end if;
  d:=iquo(k,2);
  M:=pgcd(tr(f,2*d),tr(g,2*d-(df-dg)),d);
  R:=M.Vector(2,[f,g]);
  dr:=degree(normal(R[2]));
  if R[2]=0 or k < df-dr then return M; end if;
  q:=quo(R[1],R[2],x); r:=rem(R[1],R[2],x);
  rho:=lcoeff(r); r:=r/rho; dd:=k-(df-dr);
  N:=pgcd(tr(R[2],2*dd),tr(r,2*dd-(dr-degree(normal(r)))),dd);
  return map(normal,N.Matrix(2,2,[[0,1],[1/rho,-q/rho]]).M);
end proc;
```