The Mathematica program for the algorithm is:

GCDList[{list1\_, list2\_, P\_, Q\_}] := { Drop[First[list] list1 - First[list1] list2, 1], Drop[Last[list2] list1 - Last[list1] list2, -1], (First[list2] P - First[list1] Q)/x, Last[list2] P - Last[list1] Q }

this routine doesn't test the . The variables P and Q are there just for keeping track of the linear combination of P and Q which leads to the next step.

GCDListMax[{list1\_, list2\_, P\_, Q\_}] := Module[{p1, q1}, If[Last[list2] First[list1] - Last[list1] First[list2] == 0, Return[{list1, RotateRight[First[list1] list2 - First[list2] list1], P, x ( First[list1] Q - First[list2] P)}], p1 = Drop[First[list2] list1 - First[list1] list2, 1]; q1 = Drop[Last[list2] list1 - Last[list1] list2, -1]; Return[{p1/Max[p1], q1/Max[q1], 1/Max[p1]/x (First[list2] P - First[list1] Q), 1/Max[q1] (Last[list2] P - Last[list1] Q)}]]]

this routine tests  and uses the ``max" to normalize the coefficients at each step.