G25:=GL(2,Integers(25));

G5:=GL(2,Integers(5));

S5:=sub<G5|[[2,0,0,1],[1,0,0,2]]>;

LS5:=sub<G25|[[2,0,0,1],[1,0,0,2],[1+5,0,0,1],[1,0,0,1+5],[1,5,0,1],[1,0,5,1]]>;

L:=Subgroups(LS5);

L:=<H`subgroup : H in L>;

L2:=<>;

for i:=1 to #L do

 G:=L[i];

 Com:=CommutatorSubgroup(G);

 V:=RSpace(G);

 vecs:=[v: v in V | 5\*v ne V!0];

 notyet:=true;

 for v in vecs do

 if notyet then

 U:=sub<V|v>;

 Comv := Stabiliser(Com,U);

 if Comv eq Com then

 notyet:=false;

 end if;

 end if;

 end for;

 if notyet eq false then

 L2:=Append(L2,<G,Com>);

 end if;

end for;

m:=25;

L3:=<>;

for pair in L2 do

 H:=pair[1];

 dets:={Determinant(g) : g in H};

 dets2:={k: k in [0..m-1] | GCD(k,m) eq 1};

 if dets eq dets2 then

 L3:=Append(L3,pair);

 end if;

end for;

;

ppowerisog := function(G)

 m:=Characteristic(BaseRing(G));

 H:=sub<GL(2,Integers(m))|Generators(G)>;

 V := RSpace(H);

 Odds:=[[i,j] : i in [0..m-1], j in [0..m-1]];

 stables:=[];

 for k in [1..#Odds] do

 U:=sub<V|V!Odds[k]>;

 Hv := Stabiliser(H,U);

 if Hv eq H then

 stables:=Append(stables,U);

 end if;

 end for;

 stableset:={stables[k]: k in [1..#stables]};

 stablelist:=[U: U in stableset];

 isog:=[];

 for i:=1 to #stablelist do

 isog:=Append(isog,#stablelist[i]);

 end for;

 isogset:={isog[i]: i in [1..#isog]};

 return isogset;

end function;

;

L4:=<>;

for pair in L3 do

 isog:=ppowerisog(pair[1]);

 if 25 notin isog then

 L4:=Append(L4,pair);

 end if;

end for;

;