The group acting on the Devil's Ton:

```
gap> perms := [ (2,6,11,15,20)(3,7,12,16,21),
>               (4,8,13,17,22)(5,9,14,18,23),
>               (3,8,12,17,21)(4,9,13,18,22),
>               (1,6,10,15,19)(2,7,11,16,20) ];;

gap> P := Group( perms, () );;

gap> P.name := "P";;

gap> Size( P );
12926008369442488320000

gap> Collected( Factors( Size( P ) ) );
[ [ 2, 18 ], [ 3, 9 ], [ 5, 4 ], [ 7, 3 ], [ 11, 2 ],
  [ 13, 1 ], [ 17, 1 ], [ 19, 1 ], [ 23, 1 ] ]

gap> DisplayCompositionSeries( P );
P (size 12926008369442488320000)
 | A(23)
<1> (0 gens)
```

Consider the stabilizer of the set [1,..,5] in P:

```
gap> S := Stabilizer( P, [1..5], OnSets );
Subgroup( P,
 [ ( 6,19,18,20)( 8,10,15, 9),
   ( 8, 9)(19,20),
   ( 6,18,20,22)( 8, 9),
   ( 6,22,20, 8,10)( 9,18)(15,19),
   ( 6,21,22,18, 9),
   ( 8,16,19,18)(10,12),
   ( 6,20,14,16, 7,22,18)(10,12,17,13,19,15,21),
   (18,22,23),
   ( 6, 7,11,18,17,21,19,10,12,23,22, 9)( 8,13),
   ( 4, 5)( 6,22,15, 8,20,19,18)( 9,10),
   ( 3, 4)( 6, 8,22, 9,15,21)(10,18)
   (19,20),
   ( 2, 3)( 6,10,14,16,19)( 8,23,15,22,21,12),
   ( 1, 2)( 6,19,18,16, 8, 7)( 9,23,14,12,20,10)(15,22) ] )

gap> DisplayCompositionSeries( S );
<G> (13 gens, size 384142422343680000)
 | Z(2)
<S> (12 gens, size 192071211171840000)
 | A(5) ~ ...
<S> (9 gens, size 3201186852864000)
 | A(18)
<1> (0 gens)
```

Consider the normalizer of its Sylow 5-subgroup:

```
gap> U := SylowSubgroup( S, 5 );
Subgroup( P,
 [ ( 1, 3, 5, 2, 4)( 8,11,12,23,16),
   ( 1, 4, 2, 5, 3)( 7,13,17,15,21),
   ( 1, 5, 4, 3, 2)( 7,17,21,13,15)
   ( 8,11,12,23,16)( 9,10,18,22,20),
   ( 1, 4, 2, 5, 3) ] )

gap> Size(U);
625

gap> N := Normalizer( S, U );;

gap> Collected( Factors( Size( N ) ) );
[ [ 2, 9 ], [ 3, 2 ], [ 5, 4 ] ]

gap> IsSolvable(N);
true

gap> Centre(N);
Subgroup( P, [  ] )

gap> Syl := List( [2,3,5], x-> SylowSubgroup(N, x ) );;

gap> List(Syl, x-> IsNormal(N,x));
[ false, false, true ]
```

```
gap> M := Normalizer( N, Syl[1] );;

gap> Collected( Factors( Size( M ) ) );
[ [ 2, 9 ] ]

gap> M := Normalizer( N, Syl[2] );;

gap> Collected( Factors( Size( M ) ) );
[ [ 2, 5 ], [ 3, 2 ], [ 5, 2 ] ]

gap> M := Centralizer( N, Syl[3] );

gap> Collected( Factors( Size( M ) ) );
[ [ 3, 1 ], [ 5, 4 ] ]
```

Create a permutation group of degree 60:

```
gap> A := AlternatingGroup( 5 );
Group( (1,2,5), (2,3,5), (3,4,5) )

gap> B := Operation( A, Elements( A ) );;

gap> orb := Orbits( B, [1..60] );
[ [ 1 ],
  [ 2, 29, 11, 5, 53, 41, 7, 6, 3, 8, 38, 20, 30, 52,
    40, 10, 50, 23, 16, 25 ],
  [ 4, 42, 9, 31, 48, 54, 28, 12, 60, 44, 56, 14, 13, 34,
    15 ],
  [ 17, 24, 27, 19, 36, 35, 43, 58, 39, 49, 45, 59 ],
  [ 18, 46, 33, 21, 32, 47, 57, 55, 22, 26, 51, 37 ] ]

gap> O := Operation( B, orb[2] );;

gap> Size(O);
60

gap> bl := Blocks( O, [1..20] );
[ [ 1, 5, 13, 15 ], [ 2, 7, 11, 19 ],
  [ 3, 6, 12, 17 ], [ 4, 9, 10, 16 ],
  [ 8, 14, 18, 20 ] ]

gap> P := Operation( O, bl, OnSets );
Group( (1,2,4), (1,3,2), (2,3,5) )

gap> Size( P );
60
```

Create a permutation group of degree 5^5 which cannot be reduced:

```
gap> G := GL( 5, 5 );
GL(5,5)

gap> V := GF(5)^5;
RowSpace( GF(5),
[ [ Z(5)^0, 0*Z(5), 0*Z(5), 0*Z(5), 0*Z(5) ],
  [ 0*Z(5), Z(5)^0, 0*Z(5), 0*Z(5), 0*Z(5) ],
  [ 0*Z(5), 0*Z(5), Z(5)^0, 0*Z(5), 0*Z(5) ],
  [ 0*Z(5), 0*Z(5), 0*Z(5), Z(5)^0, 0*Z(5) ],
  [ 0*Z(5), 0*Z(5), 0*Z(5), 0*Z(5), Z(5)^0 ] ] )

gap> Size(V);
3125

gap> O := Operation( G, Elements(V), OnRight );;

gap> StabChainOptions.random := 100;
100

gap> Size(O);
226614960000000000

gap> time;
41170

gap> Size(G);
226614960000000000
```

Define a finitely presented group:

```
gap> F := FreeGroup( 2 );
Group( f.1, f.2 )

gap> f := F.generators;
[ f.1, f.2 ]

gap> rels := [f[1]^3 / (f[1]*f[2])^4,
               Comm(f[1], f[2])^3,
               (f[1]^f[2])^9];
[ f.1^3*f.2^-1*f.1^-1*f.2^-1*f.1^-1*f.2^-1*f.1^-1*\
  f.2^-1*f.1^-1,
  f.1^-1*f.2^-1*f.1*f.2*f.1^-1*f.2^-1*f.1*f.2*f.1^-1*\
  f.2^-1*f.1*f.2,
  f.2^-1*f.1^9*f.2 ]

gap> H := F / rels;
Group( f.1, f.2 )
```

AbelianInvariants is the starting point.

```
gap> AbelianInvariants(H);
[ 4, 9 ]
```

Then we can try to get the order.

```
gap> Size(H);
Error, the coset enumeration has defined more
  than 64000 cosets:
type 'return;' if you want to continue with a
  new limit of 128000 cosets,
type 'quit;' if you want to quit the coset
  enumeration,
type 'maxlimit := 0; return;' in order to continue
  without a limit, in
AugmentedCosetTableMtc( G, H, -1, "_x" ) called from
D.operations.Size( D ) called from
Size( H ) called from
main loop
brk> quit;
```

So use LowIndexSubgroup.

```
gap> U := TrivialSubgroup( H );

gap> sub := LowIndexSubgroupsFpGroup( H, U,10 );;

gap> Length(sub);
74
```

And compute abelian invariants for all subgroups.

```
gap> List( sub, AbelianInvariants );
[ [ 4, 9 ], [ 2, 3, 9 ], [ 3, 3, 3, 9 ],
  [ 3, 4, 9 ], [ 0, 3, 3, 9 ], [ 2, 2, 3, 3, 9 ],
  [ 2, 3, 3, 3, 3, 9 ], [ 2, 2, 3, 3, 9 ], [ 0, 3, 9 ],
  [ 0, 3, 9 ], [ 2, 4, 9 ], [ 0, 3, 9 ], [ 0, 3, 9 ],
  [ 2, 4, 9 ], [ 2, 3, 3, 3, 9 ], [ 2, 3, 3, 9 ],
  [ 0, 0, 3, 9 ], [ 0, 9 ], [ 4, 4, 9 ], [ 0, 2, 3, 9 ],
  [ 0, 3, 4, 9 ], [ 0, 3, 9 ], [ 2, 3, 4, 9 ],
  [ 0, 0, 2, 9 ], [ 0, 9 ], [ 3, 3, 4, 9 ], [ 2, 4, 9 ],
  [ 4, 4, 9 ], [ 4, 4, 9 ], [ 2, 2, 3, 4, 9 ],
  [ 2, 4, 9 ], [ 2, 2, 2, 3, 9 ], [ 2, 2, 2, 3, 9 ],
  [ 2, 3, 4, 9 ], [ 0, 0, 2, 9 ], [ 2, 4, 9 ],
  [ 2, 4, 9 ], [ 2, 4, 9 ], [ 3, 3, 4, 9 ], [ 0, 3, 9 ],
  [ 3, 3, 4, 9 ], [ 3, 3, 4, 9 ], [ 0, 0, 4, 9 ],
  [ 0, 2, 3, 9 ], [ 2, 3, 4, 9 ], [ 0, 2, 3, 4 ],
  [ 0, 2, 2, 3, 4 ], [ 0, 2, 3, 4, 4 ],
  [ 0, 2, 3, 4, 4 ], [ 0, 2, 2, 2, 3, 4 ],
  [ 0, 3, 4, 4 ], [ 0, 3, 4, 4 ], [ 0, 3, 3, 3, 4 ],
  [ 0, 0, 2, 2, 3, 4 ], [ 3, 3, 4, 9 ], [ 0, 0, 2, 3 ],
  [ 0, 0, 3, 3, 4 ], [ 0, 2, 4 ], [ 0, 3, 4, 4 ],
  [ 0, 3, 3, 3, 4 ], [ 0, 0, 2, 2, 3, 4 ], [ 2, 3, 4, 9 ],
  [ 3, 3, 4, 9 ], [ 0, 0, 2, 3 ], [ 0, 2, 4 ], [ 3, 4 ],
  [ 2, 2, 3, 3, 4 ], [ 0, 2, 3, 4, 4 ], [ 0, 2, 2, 2, 3, 4 ],
  [ 4 ], [ 0, 2, 4 ], [ 0, 2, 3, 3, 3 ], [ 2, 3, 3, 3 ],
  [ 0, 0, 2, 2, 2, 3 ] ]
```

Therefore H in infinte!

Another group:

```
gap> F;
Group( F1, F3 )

gap> F.relators;
[ F1^3, F3^3, F1*F3*F1^-1*F3*F1*F3*F1^-1*F3,
  F3^-1*F1^-1*F3*F1^-1*F3^-1*F1^-1*F3*F1^-1,
  F3^-1*F1*F3*F1^-1*F3^-1*F1*F3*F1^-1*F3^-1*F1*F3*F1^-1 ]

gap> Size(F);
60

gap> P := OperationCosetsFpGroup( F, TrivialSubgroup( F ) );
Group( ( 1, 2, 4)( 3, 6,11)( 5, 9,16)( 7,13,22)( 8,14,24)
(10,18,30)(12,20,34)(15,26,42)(17,28,45)(19,32,48)(21,35,50)
(23,37,53)(25,40,52)(27,38,54)(29,33,49)(31,36,51)(39,47,57)
(41,44,55)(43,46,56)(58,59,60),
( 1, 3, 7)( 2, 5,10)( 4, 8,15)( 6,12,21)( 9,17,29)(11,19,33)
(13,23,38)(14,25,41)(16,27,44)(18,31,47)(20,28,40)(22,36,52)
(24,39,35)(26,43,32)(30,46,34)(37,45,42)(48,57,54)(49,58,51)
(50,59,53)(55,60,56) )
```

The other way around:

```
gap> M := MathieuGroup(12);
Group( ( 1, 2, 3, 4, 5, 6, 7, 8, 9,10,11),
( 3, 7,11, 8)( 4,10, 5, 6), ( 1,12)
( 2,11)( 3, 6)( 4, 8)( 5, 9)( 7,10) )

gap> F := FpGroup(M);
Group( F1, F2, F3 )

gap> Length(F.relators);
196

gap> P := PresentationFpGroup( F );
<< presentation with 3 gens and 194 rels of total length 87425 >>
gap> SimplifyPresentation(P);
#I  there are 3 generators and 187 relators of total length 21100
gap> SimplifyPresentation(P);
#I  there are 3 generators and 187 relators of total length 17374
gap> SimplifyPresentation(P);
#I  there are 3 generators and 186 relators of total length 14903
gap> SimplifyPresentation(P);
#I  there are 3 generators and 186 relators of total length 13742
gap> SimplifyPresentation(P);
#I  there are 3 generators and 186 relators of total length 12962
gap> SimplifyPresentation(P);
#I  there are 3 generators and 186 relators of total length 12507

gap> H := FpGroupPresentation( P );
Group( F1, F2, F3 )
```

Write an element of M in the generators of M:

```
gap> g := Random(M);
( 1,11, 2, 8, 5, 7, 9,10)( 4,12)

gap> PreImagesRepresentative( F.bijection, g );
F1^-1*F3^-1*F1^-2*F3^-1*F1^-2*F3^-1*F1^-2*F2^-2*\
F1^-3*F3^-1*F1^-2*F3^-1*F1^-2*F3^-1*F1^-2*F2^-2*\
F1^-2*F2^-3*F1^-1*F3^-1*F1^-2*F3^-1*F1^-2*F3^-1*\
F1^-2*F3^-1*F1^-2*F3^-1*F1^-2*F3^-1*F1^-2*F3^-1*\
F1^-2*F3^-1*F1^-1*F3^-1*F2^-1*F3^-1*F1^-1*F3^-1*\
F1^-2*F3^-1*F1^-2*F3^-1*F1^-2*F3^-1*F1^-2*F3^-1*\
F1^-2*F3^-1*F1^-2*F3^-1*F1^-2*F3^-1*F1^-1*F3^-1*\
F2^-1*F3^-1*F1^-1*F3^-1*F1^-2*F3^-1*F1^-2*F3^-1*\
F1^-2*F3^-1*F1^-2*F3^-1*F1^-2*F3^-1*F1^-2*F3^-1*\
F1^-2*F3^-1*F1^-1*F3^-1*F2^-1*F3^-1*F1^-1*F3^-1*\
F1^-2*F3^-1*F1^-2*F3^-1*F1^-2*F2^-2*F1^-2*F2^-2*\
F1^-1*F3^-1*F1^-2*F3^-1*F1^-2*F3^-1*F2^-1*F1^-2*\
F3^-1*F1^-2*F3^-1*F1^-2*F3^-1*F1^-2*F2^-2*F1^-3*\
F3^-1*F1^-2*F3^-1*F1^-2*F3^-1*F1^-2*F2^-2*F1^-2*\
F2^-3*F1^-1*F3^-1*F1^-2*F3^-1*F1^-2*F3^-1*F1^-2*\
F3^-1*F1^-2*F3^-1*F1^-2*F3^-1*F1^-2*F3^-1*F1^-2*\
F3^-1*F1^-1*F3^-1*F2^-1*F3^-1*F1^-1*F3^-1*F1^-2*\
F3^-1*F1^-2*F3^-1*F1^-2*F3^-1*F1^-2*F3^-1*F1^-2*\
F3^-1*F1^-2*F3^-1*F1^-2*F3^-1*F1^-1*F3^-1*F2^-1*\
F3^-1*F1^-1*F3^-1*F1^-2*F3^-1*F1^-2*F3^-1*F1^-2*\
F3^-1*F1^-2*F3^-1*F1^-2*F3^-1*F1^-2*F3^-1*F1^-2*\
F3^-1*F1^-1*F3^-1*F2^-1*F3^-1*F1^-1*F3^-1*F1^-2*\
F3^-1*F1^-2*F3^-1*F1^-2*F2^-2*F1^-2*F2^-2*F1^-1*\
F3^-1*F1^-2*F3^-1*F1^-2*F3^-1*F2^-1*F1^-2*F3^-1*\
F1^-2*F3^-1*F1^-2*F3^-1*F1^-2*F2^-2*F1^-3*F3^-1*\
F1^-2*F3^-1*F1^-2*F3^-1*F1^-2*F2^-2*F1^-3*F3^-1*\
```

```
F1^-2*F3^-1*F1^-2*F3^-1*F1^-2*F2^-2*F1^-2*F2^-3*\
F1^-1*F3^-1*F1^-2*F3^-1*F1^-2*F3^-1*F1^-2*F3^-1*\
F1^-2*F3^-1*F1^-2*F3^-1*F1^-2*F3^-1*F1^-2*F3^-1*\
F1^-1*F3^-1*F2^-1*F3^-1*F1^-1*F3^-1*F1^-2*F3^-1*\
F1^-2*F3^-1*F1^-2*F3^-1*F1^-2*F3^-1*F1^-2*F3^-1*\
F1^-2*F3^-1*F1^-2*F3^-1*F1^-1*F3^-1*F2^-1*F3^-1*\
F1^-1*F3^-1*F1^-2*F3^-1*F1^-2*F3^-1*F1^-2*F3^-1*\
F1^-2*F3^-1*F1^-2*F3^-1*F1^-2*F3^-1*F1^-2*F3^-1*\
F1^-1*F3^-1*F2^-1*F3^-1*F2^-3*F1^-1*F3^-1*F1^-2*\
F3^-1*F1^-2*F3^-1*F1^-2*F3^-1*F1^-2*F3^-1*F1^-2*\
F3^-1*F1^-2*F3^-1*F1^-2*F3^-1*F1^-1*F3^-1*F2^-1*\
F3^-1*F1^-1*F3^-1*F1^-2*F3^-1*F1^-2*F3^-1*F1^-2*\
F3^-1*F1^-2*F3^-1*F1^-2
```

Create a matrix group over GF(3).

```
gap> PrintArray(mats[1]);
[ [  0*Z(3),  Z(3)^0,  0*Z(3),  0*Z(3) ],
  [  Z(3)^0,  Z(3)^0,  0*Z(3),  0*Z(3) ],
  [  0*Z(3),  0*Z(3),  0*Z(3),  Z(3)^0 ],
  [  0*Z(3),  0*Z(3),    Z(3),  Z(3)^0 ] ]

gap> PrintArray(mats[2]);
[ [  0*Z(3),  0*Z(3),  Z(3)^0,  0*Z(3) ],
  [  0*Z(3),  0*Z(3),  0*Z(3),  Z(3)^0 ],
  [  Z(3)^0,  0*Z(3),  0*Z(3),  0*Z(3) ],
  [  0*Z(3),  Z(3)^0,  0*Z(3),  0*Z(3) ] ]
```

Find out whether it acts irreducible.

```
gap> M := GModule( mats );;

gap> IsIrreducible( M );
true
```

Use matrix group recognition.

```
gap> U := Group( mats, mats[1]^0 );;

gap> R := RecogniseMatrixGroup( U );;

gap> DisplayMatRecord( last );
#I  Matrix group over field GF(3) of dimension 4 has size 4608
#I  Number of layers is 4

gap> DisplayMatRecord( last, 1 );
#I  Layer Number = 1
#I  Type = Imprimitive
#I  Dimension = 4
#I  Size = 2

gap> DisplayMatRecord( last, 2 );
#I  Layer Number = 2
#I  Type = SL
#I  Dimension = 2
#I  Size = 48

gap> DisplayMatRecord( last, 3 );
#I  Layer Number = 3
#I  Type = SL
#I  Dimension = 2
#I  Size = 48

gap> DisplayMatRecord( last, 4 );
#I  Layer Number = 4
#I  Type = PGroup
#I  Dimension = 4
#I  Size = 1
```

Crystallographic groups

```
gap> RequirePackage("cryst");

gap> G := CrystGroup( 3, 100 );
CrystGroup( 3, 4, 5, 1, 5 )

gap> G.generators[1];
  [ [ 1,   0, 0, 0 ],
    [ 0,  -1, 0, 0 ],
    [ 0,   0, 1, 0 ],
    [ 1/2, 0, 0, 1 ] ]

gap> P := PointGroup( G );
Group( [ [ 1, 0, 0 ], [ 0, -1, 0 ], [ 0, 0, 1 ] ],
       [ [ 0, 1, 0 ], [ -1, 0, 0 ], [ 0, 0, 1 ] ],
       [ [ -1, 0, 0 ], [ 0, -1, 0 ], [ 0, 0, 1 ] ] )

gap> Size(P);
8

gap> F := FpGroup( G );
Group( g1, g2, g3, t1, t2, t3 )

gap> F.relators;
[ g1^2*g3^-1*g2^-1*t2,
  g2^2*t2^-1,
  g3^2*t1^-1,
  g2^-1*g1^-1*g2*g1*g3^-1*g2^-1*t1^-1*t2,
  g3^-1*g1^-1*g3*g1*g3^-1*g2^-1*t1*t2,
  g3^-1*g2^-1*g3*g2*t1*t2^-1,
  t1^-1*g1^-1*t1*g1*t1*t2^-1,
```

```
  t2^-1*g1^-1*t2*g1*t1*t2,
  t3^-1*g1^-1*t3*g1,
  t1^-1*g2^-1*t1*g2*t1^2,
  t2^-1*g2^-1*t2*g2,
  t3^-1*g2^-1*t3*g2,
  t1^-1*g3^-1*t1*g3,
  t2^-1*g3^-1*t2*g3*t2^2,
  t3^-1*g3^-1*t3*g3,
  t2^-1*t1^-1*t2*t1,
  t3^-1*t1^-1*t3*t1,
  t3^-1*t2^-1*t3*t2 ]

gap> U := Subgroup( G, G.generators{[1]} );;

gap> Size(U);
"infinity"

gap> V := Subgroup( G, G.generators{[2]} );
Subgroup( CrystGroup( 3, 4, 5, 1, 5 ),
[ [ [ 0, 1, 0, 0 ],
    [ -1, 0, 0, 0 ],
    [ 0, 0, 1, 0 ],
    [ 1/2, 0, 0, 1 ] ] ] )

gap> Size(V);
4

gap> Intersection( U, V );
Subgroup( CrystGroup( 3, 4, 5, 1, 5 ), [  ] )

gap> Size(last);
1
```