HOW TO READ AND USE THE DATA FILES

1. Contents.

Each folder contains 4 files:

- `hn.g`: to construct the coset table and test it on the defining relations of H.
- `hncentre.g`: to compute the basis of the center, the Gram matrix and its determinant, and the dual basis as the inverse of the Gram matrix.
- `hnchars.g`: to evaluate the irreducible characters of H on the dual basis and thus verify the basis of the center in terms of class polynomials.
- `run.g`: to execute the above three files in that order and record the results in a log file.

2. Computing in H.

After loading the first file `hn.g` into a GAP session, one can compute with elements of H expressed as vectors over H' as follows. The GAP code examples are for $H(G_4)$.

- `null`, `one`, `u`, `u1` are the elements 0, 1, u, and u^{-1} of H', where u is the reflection generating the parabolic subgroup W', identified with the corresponding element of H.

```
gap> u1;
-bc^-1+(-ac^-1)*s+(c^-1)*s^2
```

- `basicVec(i, c)` constructs the element $cx_i \in H$, where $c \in H'$.

```
gap> x3:= basicVec(3, one);
[ 0, 0, 1, 0, 0, 0, 0, 0 ]
```

- Such vectors can be added and multiplied with scalars:

```
gap> y:= x3 + basicVec(2, u);
[ 0, +(1)*s, 1, 0, 0, 0, 0, 0 ]
gap> u1 * y;
[ 0, 1, -bc^-1+(-ac^-1)*s+(c^-1)*s^2, 0, 0, 0, 0, 0 ]
```

- `vecUnderWord(vec, word)` computes the image of a vector `vec` under a word `word` in the generators of H and their inverses, represented as a list with entries i for the generator σ_i and $-i$ for its inverse.

```
gap> x5:= vecUnderWord(x3, [1,2,1]);
[ 0, 0, 0, 0, 1, 0, 0, 0 ]
gap> vecUnderWord(x5, -[1,2,1]);
[ 0, 0, 1, 0, 0, 0, 0, 0 ]
```

- In particular, the representation of a word w can be computed as the image of x_1 under w:

```
gap> x1:= basicVec(1, one);
[ 1, 0, 0, 0, 0, 0, 0, 0 ]
gap> pi:= vecUnderWord(x1, [1,2,1,2,1,2,1,2,1,2,1,2]);
[ c^4+(bc^3)*s+(ac^3)*s^2, bc^3+(b^2c^2)*s+(abc^2)*s^2,
  +(bc^2)*s+(b^2c)*s^2, bc^2+(b^2c)*s+(abc)*s^2,
  +(-a^2c)*s+(b^2)*s^2, +(ac)*s, +(-a^2)*s+(a)*s^2, +(a)*s ]
```

- $\tau(h)$, the coefficient at b_1 can be found as h`[1].list[1]`:

```
gap> pi[1].list[1];
c^4
```

3. The Center and the Dual Basis.

Then, after loading the second file `hncentre.g` into GAP, a basis for $Z(H)$, the gram matrix $\tau(b_i b_j)$ and the dual basis $\{b_i^\vee\}$ are available as follows. Note that it can take a very long time to compute the Gram matrix.

- `coeffsVec(vec)` converts a vector `vec` over H' into a vector over R.

```
gap> x8:= basicVec(8, one);
[ 0, 0, 0, 0, 0, 0, 0, 1 ]
gap> b29:= coeffsVec(x8);
[ 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
  0, 1, 0, 0 ]
```

- `vecCoeffs(coeffs)` converts a vector `coeffs` over R into a vector over H'.

```
gap> vecCoeffs(b29);
[ 0, 0, 0, 0, 0, 0, 0, 1 ]
```

- The list `nul2` contains the basis of $Z(H)$ as vectors over R.

```
gap> nul2[1];
[ 0, 0, ac^3, 0, ac^3, abc^2+c^3, abc^2+c^3,
  bc^2+ab^2c-a^2c^2, ac^2, 0, ac^2, abc+c^2, 0,
  2bc+ab^2-a^2c, 0, 0, 0, c, c, b, 0, 0, 0, 1 ]
gap> vecCoeffs(nul2[1]);
[ +(ac^3)*s^2, +(ac^3)*s+(abc^2+c^3)*s^2,
  abc^2+c^3+(bc^2+ab^2c-a^2c^2)*s+(ac^2)*s^2,
  +(ac^2)*s+(abc+c^2)*s^2, +(2bc+ab^2-a^2c)*s, +(c)*s^2,
  c+(b)*s, +(1)*s^2 ]
```

- `gram` is the Gram matrix $\tau(b_i b_j)$:

```
gap> gram{[1..3]}{[1..3]};
[ [ 1, 0, 0 ], [ 0, 0, c ], [ 0, c, ac ] ]
```

- The list `dual` contains the dual basis as vectors over R:

```
gap> dual[2];
[ 0, -ac^-1, c^-1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
  0, 0, 0, 0, 0, 0, 0 ]
gap> vecCoeffs(dual[2]);
[ +(-ac^-1)*s+(c^-1)*s^2, 0, 0, 0, 0, 0, 0, 0 ]
```