Some `realex` functions and commands

`<file`: Read in the file called "file.rx"

: (vec->int) : number of components of vector

: (mat->int,int) : dimensions (rows*columns) of a matrix

: ([T]->int) : number of components of row (T is any type)

`Cartan_class:` (RealForm,int->CartanClass): Cartan class selected by number. This selects a Cartan class by number in the list of Cartan classes defined for this real form. The numbering is not the same as when selecting a Cartan class directly from an inner class, unless the real form is quasisplit.

`Cartan_list`: (RealForm->[CartanClass]) {workshop.rx}

`c_form_irreducible`: (Param->ParamPol) {hermitian.rx}

`c_form_std`: (Param->ParamPol) {hermitian.rx}

`character_formula:` (Param->ParamPol): writes an irreducible as a formal sum of standards {KL.rx}

`composition_series`: (Param->ParamPol): write a standard module as a formal sum of irreducibles {KL.rx}

`deform`: (Param->ParamPol): compute deformation terms when nu decreases. The non-integral block for the parameter and its KL polynomials are computed, from which the deformation terms involving other members of the block are computed. They are returned as a formal sum of parameters with split integer coefficients, which are in fact integer multiples of (1-s).

`describe_Cartan`(KGBElt->) {workshop.rx} `full_deform`: (Param->ParamPol): perform deformation all the way to nu=0. This is like deform, but recursively deforms all new terms produced as long as they do not have nu=0; all terms in the reult therefore have nu=0

`get_n_block`: (Param->[Param]) {misc.rx}

`hermitian_form_irreducible`: (Param->ParamPol) {hermitian.rx}

`infinitesimal_character`: (Param->ratvec)

`is_final`: (Param->bool)

`is_unitary`: (Param->bool): decides whether the irreducible representation is unitary (Warning: at this point, this gives the correct answer for equal rank groups only!) {hermitian.rx}

`Lie_type`: (RealForm->LieType)

`list_cartans`: (RealForm->) {workshop.rx}

`KGB`: (RealForm,int->KGBElt)

`KGB`: (RealForm->[KGBElt]) {basic.rx}

`most_split_Cartan`: (RealForm->CartanClass): most split Cartan class for form

`n_block`: (Param->[Param],int): return block as list of parameters, and index. The second component is the index into the first of the original parameter.

`param`: (KGBElt,vec,ratvec->Param): form parameter from (x,lambda-rho,nu)

`param`: (RealForm,int,vec,ratvec->Param) `print_Cartan_info`: (CartanClass->): print information about the Cartan class. This produces essentially the output of 'cartan' in the Atlas, except for the final partition corresponding to the real forms for this Cartan class. So it prints the number of split (GL(1,R)), compact (U(1)) and complex (GL(1,C)) factors of the real torus defined by this Cartan class, the number of distinct twisted involutions defining this same Cartan class, and the types of the imaginary, real, and complex root subsystems

`print_character_formula`: (Param->)   {KL.rx}

`print_composition_series`: (Param->)   {KL.rx}

`print_hermitian_form_irreducible`: (Param->)   {KL.rx}

`print_KGB`: (RealForm->):

`print_n_block`: (Param->): print (nonintegral) block generated from parameter

`print_real_Weyl`: (RealForm,CartanClass->)

`reducibility_points`: (Param->[rat]): the 0<t<=1 with I(x,lambda,t\nu) reducible

`rho`: (RealForm->ratvec)   {misc.rx}

`showall`: prints all functions and operations

`test_line`: (Param->)   {hermitian.rx}

`trivial`: (RealForm->Param)   {misc.rx}

Groups (real forms) that have been defined:

```
SU(p,q), PSU(p,q), SU(p), PSU(p)
U(p,q)
SL(n,D), GL(n,D), PSL(n,D), PGL(n,D) for D=R, C, H
```

`Sp(p,q), PSp(p,q)`
`Sp(n,D), PSp(n,D), GSp(n,D) for D=R, C; n` must be even.

```
SO(p,q), Spin(p,q), PSO(p,q)
E6_c, E6_h, E6_D5T, E6_q
E6_F4, E6_s, E6_C4

E7_c, E7_h, E7_E6T, E7_q, E7_D6A1, E7_q, E7_D6A1,
E7_s, E7_A7

E8_c, E8_q, E8_s

F4_c, F4_B4, F4_s

G2_c, G2_s
```