Function: Octonion:-oversion **-** display information about the current version of the 'Octonion' package

Calling Sequence:

oversion();

Parameters:

no parameters needed

Description:

- Procedure 'oversion' displays information about the current version of the 'Octonion' package.
- The 'Octonion' package must be loaded after the 'CLIFFORD' package has been loaded. Therefore, in order to avoid confusion with the procedure Clifford:-version, this procedure is called 'oversion'.
- To display 'CLIFFORD' and 'Octonion' environmental variables, use procedure Clifford:-CLIFFORD_ENV.
- To multiply octonionic matrices, see Clifford:-rmulm.

Examples:

```
> restart:with(Clifford):with(Octonion);
```

$[\Phi, associator, commutator, def\_omultable, o\_conjug, oinv, omul, omultable, onorm, oversion, purevectorpart, realpart]$

```
> version(); #current version of CLIFFORD
```

+++

CLIFFORD - A Maple 12 Package for Clifford Algebras with "Bigebra"
(Version 12 with environmental variables given by CLIFFORD_ENV())
Last revised: December 20, 2009 (Source file: clifford_M12_12.mws)
Copyright 1995-2009 by Rafal Ablamowicz (*) and Bertfried Fauser ($)

(*) Department of Mathematics, Box 5054
Tennessee Technological University, Cookeville, TN 38505
tel: USA (931) 372-3569, fax: USA (931) 372-6353
rablamowicz@tntech.edu
http://math.tntech.edu/rafal/
($) Universit"at Konstanz, Fachbereich Physik, Fach M678
78457 Konstanz, Germany
Bertfried.Fauser@uni-konstanz.de
http://kaluza.physik.uni-konstanz.de/~fauser/

If you are a Clifford algebra pro, assign 'true' to '_prolevel' and see

how much faster your computations will be! But watch your syntax!

Use 'useproduct' to change value of _default_Clifford_product in Cl(B) from

cmulRS when B is symbolic to cmulNUM when B is numeric. Type ?cmul for help.

Type CLIFFORD_ENV() to see current values of environmental variables.

++++++++++++This is CLIFFORD version 12++++++++++++

```
> oversion(); #current version of Octonion
```

++

'Octonion' - A Maple 12 Package for Computations with Octonions (version 12)

Last revised: December 20, 2008

Copyright 1995-2009, by Rafal Ablamowicz, Tennessee Technological University

Department of Mathematics, Box 5054

Tennessee Technological University, Cookeville, TN 38505

phone: USA (931) 372-3569, fax: USA (931) 372-6353

e-mail: rablamowicz@tntech.edu

http://math.tntech.edu/rafal/

++

```
>
```

See Also: omul

Function: Octonion:-setup - the initialization procedure for the package 'Octonion'

Calling Sequence:

none

Parameters:

none

Description:

- Procedure 'setup' is the initialization procedure for the 'Octonion' package. It is executed automatically when the package is loaded.
- At the time of loading, the following are defined:

  - `&o` - infix form for omul, the octonionic multiplication
  - _octbasis = [Id, e1, e2, e3, e4, e5, e6, e7] - standard octonion basis as Maple global variable in Cl(0,7)
  - _pureoctbasis = [e1, e2, e3, e4, e5, e6, e7] - pure octonion basis as Maple global variable in Cl(0,7)
  - _default_Fano_triples = [[1,3,7],[1,2,4],[1,5,6],[2,3,5],[2,6,7],[3,4,6],[4,5,7]] - default Fano triples that define octonionic multiplication
  - _default_squares = [-Id, -Id, -Id, -Id, -Id, -Id, -Id] - default squares of the pure octonionic basis

- To see all environmental raviables that are defined and used by 'CLIFFORD', use procedure Clifford:-CLIFFORD_ENV.
- All procedures and types in 'Octonion' are protected.

Examples:

```
> restart:with(Clifford):with(Octonion):
> CLIFFORD_ENV();
```

```
`>>> Global variables defined in Clifford:-setup are now available and have the
se values: <<<`
`************* Start *************`
dim_V = 9
_default_Clifford_product = Clifford:-cmulNUM
_prolevel = false
_shortcut_in_minimalideal = true
_shortcut_in_Kfield = true
_shortcut_in_spinorKbasis = true
_shortcut_in_spinorKrepr = true
_warnings_flag = true
_scalartypes = {`^`, RootOf, complex, indexed, numeric, constant, function, mat
hfunc, rational}
_quatbasis = [[Id, e3we2, e1we3, e2we1], {`Maple has assigned qi:=-e2we3, qj:=e
1we3, qk:=-e1we2`}]
```

```
`************* End *************`

Cliplus has been loaded. Definitions for type/climon and type/clipolynom now in
clude &C and &C[K]. Type ?cliprod for help.

`>>> Global variables defined in Cliplus:-setup are now available and have thes
e values:  <<<`
`************* Start *************`
macro(Clifford:-cmul = climul)
macro(Clifford:-cmulQ = climul)
macro(`&c` = climul)
macro(`&cQ` = climul)
macro(Clifford:-reversion = clirev)
macro(Clifford:-LC = LCbig)
macro(Clifford:-RC = RCbig)
`Warning, new definitions for type/climon and type/clipolynom now include &C`
`************* End *************`



`************* Start *************`
`>>> There are no new global variables or macros in GTP yet. <<<`
`************* End *************`



`>>> Global variables defined in Octonion:-setup are now available and have the
se values:  <<<`

`************* Start *************`
_octbasis = [Id, e1, e2, e3, e4, e5, e6, e7]
_pureoctbasis = [e1, e2, e3, e4, e5, e6, e7]
_default_Fano_triples = [[1, 3, 7], [1, 2, 4], [1, 5, 6], [2, 3, 5], [2, 6, 7],
[3, 4, 6], [4, 5, 7]]
_default_squares = [-Id, -Id, -Id, -Id, -Id, -Id, -Id]
_default_Clifford_product = Clifford:-cmulNUM
`************* End *************`

>
```

See Also: `type/Fano_triples`, omultable, omul

Last revised: December 20, 2008/RA/BF

Function: Octonion:-associator **-** returns the associator value of three octonions,
Octonion:-commutator **-** returns the commutator value of two octonions
Octonion:-Phi - associative 3-form of three octonions

Calling Sequence:

associator(p1,p2,p3);
commutator(p1,p2);
Phi(p1,p2,p2);

Parameters:

p1, p2, p3 - polynomials of type 'octonion'

Description:

- The associator of three octonions p1, p2, and p3 is defined as:

  associator(p1,p2,p3) = (p1 &o p2) &o p3 - p1 &o (p2 &o p3).

- The commutator of two octonions p1 and p2 is defined as:

  commutator(p1,p2) = p1 &o p2 - p2 &o p1.

- The associative 3-form Phi of three octonions is defined as:

  Phi(p1,p2,p3) = (1/2)*realpart(p1 &o (p2_bar &o p3) - p3 &o (p2_bar &o p1))

  where p2_bar = o_conjug(p2).

- For information about type 'octonion' see `type/octonion`.

Examples:

```
> restart:with(Clifford):with(Octonion);
```

$$[\Phi, associator, commutator, def\_omultable, o\_conjug, oinv, omul, omultable, onorm, oversion, purevectorpart, realpart]$$

```
> p1:=1-2*e1+e4+3*e6-e7;p2:=2-e1+e3+2*e6-e7;p3:=2*e2+e3+3*e5-e6;
```

$$p1 := 1 - 2\,e1 + e4 + 3\,e6 - e7$$
$$p2 := 2 - e1 + e3 + 2\,e6 - e7$$
$$p3 := 2\,e2 + e3 + 3\,e5 - e6$$

```
> type(p1,octonion);type(p2,octonion);type(p3,octonion);
Cliplus has been loaded. Definitions for type/climon and type/clipolynom now in
clude &C and &C[K]. Type ?cliprod for help.
```

$$true$$

$$true$$

$$true$$

Octonion multiplication is not associative:

```
> associator(p1,p2,p3);
```

$$-8\, e1 - 2\, e2 + 20\, e3 + 14\, e4 - 6\, e5 - 2\, e6 + 24\, e7$$

However, when p1, p2, and p3 are considered as elements in the Clifford algebra Cl(0,7), which is associative, we get:

```
> (p1 &c p2) &c p3 - p1 &c (p2 &c p3);
```

$$0$$

```
> commutator(p1,p2);
```

$$2\, e1 - 4\, e2 + 2\, e3 + 6\, e4 + 4\, e5 - 2\, e6 - 4\, e7$$

```
> Phi(p1,p2,p3);
```

$$4$$

```
>
>
```

See Also: Clifford:-`&c`, def_omultable, omultable, omul

Last revised: December 20, 2008/RA/BF

Function: Octonion:-associator **-** returns the associator value of three octonions,
Octonion:-commutator **-** returns the commutator value of two octonions
Octonion:-Phi - associative 3-form of three octonions

Calling Sequence:

associator(p1,p2,p3);
commutator(p1,p2);
Phi(p1,p2,p2);

Parameters:

p1, p2, p3 - polynomials of type 'octonion'

Description:

- The associator of three octonions p1, p2, and p3 is defined as:

  associator(p1,p2,p3) = (p1 &o p2) &o p3 - p1 &o (p2 &o p3).

- The commutator of two octonions p1 and p2 is defined as:

  commutator(p1,p2) = p1 &o p2 - p2 &o p1.

- The associative 3-form Phi of three octonions is defined as:

  Phi(p1,p2,p3) = (1/2)*realpart(p1 &o (p2_bar &o p3) - p3 &o (p2_bar &o p1))

  where p2_bar = o_conjug(p2).

- For information about type 'octonion' see `type/octonion`.

Examples:

```
> restart:with(Clifford):with(Octonion);
```

$[\Phi, associator, commutator, def\_omultable, o\_conjug, oinv, omul, omultable, onorm, oversion, purevectorpart, realpart]$

```
> p1:=1-2*e1+e4+3*e6-e7;p2:=2-e1+e3+2*e6-e7;p3:=2*e2+e3+3*e5-e6;
```

$$p1 := 1 - 2\ e1 + e4 + 3\ e6 - e7$$
$$p2 := 2 - e1 + e3 + 2\ e6 - e7$$
$$p3 := 2\ e2 + e3 + 3\ e5 - e6$$

```
> type(p1,octonion);type(p2,octonion);type(p3,octonion);
Cliplus has been loaded. Definitions for type/climon and type/clipolynom now in
clude &C and &C[K]. Type ?cliprod for help.
```

$$true$$
$$true$$
$$true$$

Octonion multiplication is not associative:

```
> associator(p1,p2,p3);
```

$$-8\,e1-2\,e2+20\,e3+14\,e4-6\,e5-2\,e6+24\,e7$$

However, when p1, p2, and p3 are considered as elements in the Clifford algebra Cl(0,7), which is associative, we get:

```
> (p1 &c p2) &c p3 - p1 &c (p2 &c p3);
```

$$0$$

```
> commutator(p1,p2);
```

$$2\,e1-4\,e2+2\,e3+6\,e4+4\,e5-2\,e6-4\,e7$$

```
> Phi(p1,p2,p3);
```

$$4$$

```
>
>
```

See Also: Clifford:-`&c`, def_omultable, omultable, omul

Last revised: December 20, 2008/RA/BF

Function: Octonion:-def_omultable **-** define octonionic multiplication table

Calling Sequence:

def_omultable(F);

Parameters:

F - a list of type `Fano_triples`

Description:

- Procedure 'def_omultable' allows user to define an octonionic multiplication table which could be different than the default one.
- The default multiplication table is initialized at the time when the 'OCTONION' package is being loaded. It can also be re-defined by issuing the following command:

  ```
  >def_omultable(_default_Fano_triples);
  ```

  where _default_Fano_triples is a global list with default Fano triples. See `type/Fano_triples` for more information.
- Use omultable to display currently defined multiplication table.
- Use Clifford:-CLIFFORD_ENV to display current environmental variables used by 'CLIFFORD' and 'Octonion'.

Examples:

```
> restart:with(Clifford):with(Octonion);
```

$[\Phi, associator, commutator, def\_omultable, o\_conjug, oinv, omul, omultable, onorm, oversion, purevectorpart, realpart]$

```
> omultable(); #default multiplication table
```

$$\begin{bmatrix} -Id & e4 & e7 & -e2 & e6 & -e5 & -e3 \\ -e4 & -Id & e5 & e1 & -e3 & e7 & -e6 \\ -e7 & -e5 & -Id & e6 & e2 & -e4 & e1 \\ e2 & -e1 & -e6 & -Id & e7 & e3 & -e5 \\ -e6 & e3 & -e2 & -e7 & -Id & e1 & e4 \\ e5 & -e7 & e4 & -e3 & -e1 & -Id & e2 \\ e3 & e6 & -e1 & e5 & -e4 & -e2 & -Id \end{bmatrix}$$

For example, we get the first row as follows:

```
> seq(e1 &o e||i,i=1..7);
```

$$-Id, e4, e7, -e2, e6, -e5, -e3$$

The second row we get as follows:

```
> seq(e2 &o e||i,i=1..7);
```

$$-e4, -Id, e5, e1, -e3, e7, -e6$$

and so on.

Multiplication table can be erased as follows:

```
> subsop(4=NULL,eval(omul)):
> omultable();
Octonion multiplication table is not currently defined. Use 'def_omultable' to
define a new table.
```

Finally, we re-initialize the table using the default Fano triples:

```
> _default_Fano_triples;
```

$$[[1,3,7],[1,2,4],[1,5,6],[2,3,5],[2,6,7],[3,4,6],[4,5,7]]$$

```
> def_omultable(_default_Fano_triples);
> omultable();
```

$$\begin{bmatrix} -Id & e4 & e7 & -e2 & e6 & -e5 & -e3 \\ -e4 & -Id & e5 & e1 & -e3 & e7 & -e6 \\ -e7 & -e5 & -Id & e6 & e2 & -e4 & e1 \\ e2 & -e1 & -e6 & -Id & e7 & e3 & -e5 \\ -e6 & e3 & -e2 & -e7 & -Id & e1 & e4 \\ e5 & -e7 & e4 & -e3 & -e1 & -Id & e2 \\ e3 & e6 & -e1 & e5 & -e4 & -e2 & -Id \end{bmatrix}$$

However, the following is another valid list of Fano triples:

```
> new_Fano_triples:=[[6,2,5],[6,3,4],[6,7,1],[2,3,7],[3,1,5],[2,4
,1],[4,5,7]];
```

$$new\_Fano\_triples := [[6,2,5],[6,3,4],[6,7,1],[2,3,7],[3,1,5],[2,4,1],[4,5,7]]$$

```
> type(new_Fano_triples,Fano_triples);
```

$$true$$

```
> def_omultable(new_Fano_triples);
> omultable();
```

$$\begin{bmatrix} -Id & e4 & -e5 & -e2 & e3 & e7 & -e6 \\ -e4 & -Id & e7 & e1 & e6 & -e5 & -e3 \\ e5 & -e7 & -Id & e6 & -e1 & -e4 & e2 \\ e2 & -e1 & -e6 & -Id & e7 & e3 & -e5 \\ -e3 & -e6 & e1 & -e7 & -Id & e2 & e4 \\ -e7 & e5 & e4 & -e3 & -e2 & -Id & e1 \\ e6 & e3 & -e2 & e5 & -e4 & -e1 & -Id \end{bmatrix}$$

which is a different multiplication table than before.

```
> 
> 
```

See Also: `type/Fano_triples`, omultable, omul

Last revised: December 20, 2008/RA/BF