Documentation for VoidComp.h and VoidComp.c

Steven Andrews, © 2008

See the document "LibDoc" for general information about this and other libraries.

Header

```
#ifndef __VoidComp_h
#define __VoidComp_h

int StringCmp(void *str1,void *str2);
int PointerCmp(void *ptr1,void *ptr2);
int IntCmp(void *num1,void *num2);
int FloatCmp(void *num1,void *num2);
int DoubleCmp(void *num1,void *num2);

#endif
```

Requires: `"VoidComp.h"`

Example program: `Smoldyn`

Created 3/14/02 with code moved from Set.c and queue.c. Renamed `PtrCmp` to `PointerCmp` and `FltCmp` to `FloatCmp` 1/11/08. Added `DoubleCmp` at the same time.

This is just a short collection of routines for comparing different values, where they are stored as void*'s. It is useful for both Set.c and queue.c.

Functions

`int StringCmp(void *str1,void *str2);`

Compares strings, where the `char*` type is cast as a `void*`. It returns –1 if the first string is smaller, 0 if they are equal, and 1 if the first value is larger, where these comparisons use the standard library `strcmp` routine. A `NULL` input pointer is declared less than any string, and two `NULL` pointers are equal.

`int PointerCmp(void *ptr1,void *ptr2);`

Compares pointers, where the pointer is cast as a `void*`. It returns 0 if two pointers are equal and 1 if they aren't.

`int IntCmp(void *num1,void *num2);`

Compares integers, where the `int` type is cast as a `void*`. It returns –1 if the first value is smaller, 0 if they are equal, and 1 if the first value is larger.

`int FloatCmp(void *num1,void *num2);`

Compares a pair of float type numbers, which are pointed to by `num1` and `num2`. It returns –1 if `num1<num2`, 0 if `num1=num2`, and 1 if `num1>num2`. This is a useful routine for sorted queues. Comparable routines in Set.c may be useful as well.

`int DoubleCmp(void *num1,void *num2);`

Compares a pair of double type numbers, which are pointed to by `num1` and `num2`. It returns –1 if `num1<num2`, 0 if `num1=num2`, and 1 if `num1>num2`. This is a useful routine for sorted queues.