The AllocRing class

An AllocRing is a bounded ring (circular list), each of whose elements contains a pointer to some space allocated via **new char[***some_size***]**. The entries are used cyclicly.   The size, n, of the ring is fixed at construction. After that, every nth use of the ring will reuse (or reallocate) the same space. AllocRings are needed in order to temporarily hold chunks of space that are needed transiently, but across constructor-destructor scopes. They mainly useful for storing strings containing formatted characters to print across various functions and coercions. These strings are needed across routines, so may not be deleted in any one of them, but should be recovered at some point. In other words, an AllocRing is an extremely simple minded garbage collection mechanism. The GNU C++ library used to use one AllocRing for such formatting purposes, but it is being phased out, and is now only used by obsolete functions. These days, AllocRings are probably not very useful.

Support includes:

AllocRing a(int n)   constructs an Alloc ring with n entries, all null.

    void* mem = a.alloc(sz)   moves the ring pointer to the next entry, and reuses the space if their is enough, also allocates space via new char[sz].

    int present = a.contains(void* ptr) returns true if ptr is held in one of the ring entries.

    a.clear()   deletes all space pointed to in any entry. This is called automatically upon destruction.

    a.free(void* ptr)   If ptr is one of the entries, calls delete of the pointer, and resets to entry pointer to null.