Buffer Manager
Version 2 (Updated on 02/28/17)

BufferManager(bufferSize)
Constructor for the BufferManager; allocates a hash data structure of size bufferSize for buffer pool; buffer pool contains frames consisting of <pgId> as key and <pg,pinCnt,dirty,priority> as value; there is only one instance of BufferManager and a buffer pool for the entire DBMS.

readPage(pgId,policy,priority)
Returns page: if requested page is in bufferpool, returns it from main memory; otherwise, copies page from disk to buffer pool, then returns the page; if buffer pool were full, it replace a frame in buffer pool according to the policy and priority; throws an exception if requested pgId is not in the disk.

writePage(pg,pgId,policy,priority)
Checks if page is on the buffer pool. If so, update frame with pgId in the buffer pool; otherwise internally calls readPage(pgId,policy,priority) and then update the frame; set dirty bit to 1.

releasePages(pgId [], bool temp)
If temp is true, deletes pages from pool; otherwise, set priority to 0.

pin(pgId)
Increments pinCnt of a frame with pgId and returns status of operation. If pinCnt of the frame is greater than 0, then the page is “pinned”, which means that page cannot be written back to disk.

unpin(pgId)
Decrement pinCnt of a frame with pgId and returns status of operation. If pinCnt of the frame becomes 0, then the page is “unpinned” and may be written back to disk if needed.

forceOutput(pgId)
Immediately writes page from buffer pool to disk, if dirty bit is 1. Otherwise, nothing need to be written, since page in buffer is exact copy of page in disk.
flush()
Writes back all unpinned pages to disk.

iterator()
Return iterator over pages.