```
/*
* DBMS Implementation
*/

#ifndef BUFFEROPS_H
#define BUFFEROPS_H

#include <io.h>
#include <fcntl.h>
#include <sys/types.h>

#include "dbmsproj.h"
#include<iostream>


// empties a block
template<typename T> void emptyBlock(block_t<T> *buffer, unsigned int size) {

    buffer->nreserved = 0;
    buffer->entries.clear();
    //printf("empty buffer block... \n");
};

// empties the whole buffer

template<typename T> void emptyBuffer(block_t<T> *buffer, unsigned int size) {
    for (uint i = 0; i < size; i++) {
        emptyBlock(buffer + i);
        buffer[i].valid = true;
    }
    printf("empty buffer ...");
};

// opens filename for writing (append mode), and writes size blocks
// starting from pointer buffer

template<typename T> unsigned int writeBlocks(block_t<T> &relation, block_t<T> &buffer, unsigned int offset, unsigned int size) {
    unsigned int writecounts = 0;

    for (size_t i = 0; i < size; i++)
    {
        relation[i+offset] = buffer[i];
        writecounts++;
    }
    printf("Write %d blocks into disk...\n", readcount);
    return size;
};

// reads size blocks to buffer

template <typename T> unsigned int readBlocks(std::vector<block_t<T>>& relation, block_t<T>* buffer, unsigned int size) {
    unsigned int readcount = 0;
    for (unsigned int i = 0; i < size; i++)
    {
        if (i<relation.size())
        {
            buffer[i] = relation[offset + i];
            readcount++;
        }
        else
        {
            break;
        }
    }
    printf("loading %d blocks into buffer...\n",readcount);
    return readcount;
};
template <typename T> unsigned int readBlocks(std::vector<block_t<T>>& relation, block_t<T>* buffer,unsigned int size,unsigned int offset) {
    unsigned int readcount = 0;
    for (unsigned int i = 0; i < size; i++)
    {
        if ((offset+i)<relation.size())
        {
            buffer[i] = relation[offset+i];
            readcount++;
        }
        else
        {
            break;
        }

    }
    printf("loading %d blocks into buffer...\n",readcount);
    return readcount;
};
#endif
```