Compiling and running project:

Please make sure you are using linux system, this makefile don't work for mac system, contact franklyn@ufl.edu if you need makefile for other operating system.

Steps of compiling and running test.out(after entering directory of project): 1. make clean

2. make

3. make test.out

4. ./test.out

Step of running test cases script(result will be output1.txt):

1. make clean

2. make

3. ./runTestCases.sh

DBFiles are stored in bin/*bin

Two custom comparators for record and run using given OrderMaker built in BigQ

```
bool RecordComparator = [this](Record* left, Record* right)

bool RunComparator = [this](Run* left, Run* right)

priority_queue<Record*, vector<Record*>,
decltype(RecordComparator)> recordPQ(RecordComparator);

priority_queue<Run*, vector<Run*>, decltype(RunComparator)>
runPQ(RunComparator);
```

BigQ class is a data structure with input and output pipe, order and run length, and associated methods.

BigQ :: BigQ (Pipe &in, Pipe &out, OrderMaker &sortorder, int runlen);

Construct method for BigQ, it contains the data used for worker' function. Then it will create worker thread, and let worker thread to do rest of things.

Int **BigQ ::** ExcuteSortPhase()

The main thing in the worker function that execute the sort by building two priority queues.

```
class BigQ {
private:
	Pipe* input, * output;
	OrderMaker* order;
	int runlen;
	ComparisonEngine* comp;
public:
	BigQ(Pipe& in, Pipe& out, OrderMaker& sortorder, int runlen);
	~BigQ();
	int ExcuteSortPhase();
};
```

Run class is a data structure for the merge of different runs in Run Priority Queue, and associated methods.

```
class Run {
private:
	File* fileBase;
	Page bufferPage;
	int startPageIdx;
	int curPageIdx;
	int runlen;
public:
	Run(File* file, int startPageIndx, int runLength);
	int UpdateTopRecord();

    Record* topRecord;
};
```

Worker function that execute sort process via data and methods encapsulated in BigQ

```
void* Worker(void* bigQ) {
	auto* bigq = (BigQ*)bigQ;
	bigq->ExcuteSortPhase();
	return nullptr;
}
```