Project 2 part 2: Sorted File Implementation

<<<---------- -----TEAM MEMBERS---------->>>
Mohit Bharaney (UFID-49953911)
Mugdha Kumar (UFID – 54219168)

<<<---------- ----Content of Project2_2_49953911_54219168.zip---------->>>
1. a2_2test folder
2. Report.pdf

<<<---------- INTRODUCTION---------->>>
DBFile can be either heap or sorted. Implemented all the functionalities like 'Create', 'Load', 'Add', 'MoveFirst', 'Open', 'GetNext' and 'Close' for sorted file which we had already done for heap file. First we created a 'GenericDB' class which is being inherited by both 'Heap' class and 'Sorted' class. Each class (heap and sorted) has their own implementation of all the functionalities.

<<---------- STEPS FOR RUNNING---------->>>
Below are steps to compile the code and run it
1. Untar/unzip the submitted file.
2. For testing the Heap class on console compile using command **make a1test.out** and then type **./a1test.out**. Further follow the instructions on the screen.
3. For testing the Sorted class implementation under project 2 part 1 run the command **make a2test.out** and then type **./a2test.out.** Further follow the instructions on the screen.
4. For testing the Sorted class implementation under project 2 part 2 run compile using **make.** Further run ./test.out and follow the instructions on the screen.

<<---------- PROGRAMMING PRACTICES---------->>>
Run valgrind (memcheck) and lint regularly

<<---------- IMPLEMENTATION---------->>>

Created a generic class called 'GenericDB'. This class acts like a base class for heap and sorted type file.

Create function takes in file path, creates the check for the type of the file and instantiated the heap or sorted class.

Open function opens the metadata file to determine wich kind of file it is ,and instantiates the appropriate object and calls their functions.

For Sorted implementation, adds all the recs in to input pipe.

Add function , for sorted : adds records to input pipe and sets the appropriate flag and instatntiated bigq if its not started

GetNext without filter returns the next record based on the metadata file to maintain consistency, keeps track of the no of records written into a page.

GetNext with Filter gets the next record, tests it against the cnf conditions and returns 1 if true on the first comparison found.

GetNext:Sorted:-> based on the cnf and sort order a query sortorder is created which uses all the attributes in cnf that can be utilized in binary search. Then after the binary search return the page index a linear scan is done to match the cnf.

Move first moves the read pointer to the top of the file

Close writes in the file the incomplete buffer, updates the metadata file with record pointer information to maintain consistency across various sessions.