Testcase-3

Note: Default disk page size and memory frame size are 3. You can edit both in common.h. Both disk page size and memory frame sizes should always be equal.

Flow:

- Input line 1: Program reads 4 and creates main memory of 4 frames.
- Main.cpp: Line 20 calls readDiskFile() which creates a disk file by reading inputs from cin.
- Main.cpp: Line 21 calls writeDiskFile() which writes disk file on cout.
- Main.cpp: Line 26 calls twoWaySort() which takes two inputs: DiskFile object and MainMemory object which is to be used to perform sorting.
- ExtMergeSort.cpp: Line 154 checks if the number of available frames in main memory is greater than or equal to 3.
- ExtMergeSort.cpp: Line 157 calls firstPass() which loads each page into frames, sorts those frames and writes back those frames on corresponding pages of DiskFile this creates runs of 1 page size.
- ExtMergeSort.cpp: Line 167 calls merge on <0, 0, 1> which joins merges page 0 and 1.
- ExtMergeSort.cpp: Line 167 calls merge on <2, 2, 3> which joins merges page 2 and 3
- ExtMergeSort.cpp: Line 167 calls merge on <4, 4, 5> which joins merges page 4 and 5
- Now we have runs of 2 page size.
- ExtMergeSort.cpp: Line 167 calls merge on <0, 1, 3> which joins/merges run of pages <0,1> and run of pages<2,3> and creates run of pages <0,1,2,3>
- ExtMergeSort.cpp: Line 167 calls merge on <4,5,6> which joins/merges run of pages <4,5> and run of page<6> and creates run of pages <4,5,6>.
- Now we have run of maximum size 4.
- ExtMergeSort.cpp: Line167 calls merge on <0,3,6> which joins/merges run of pages <0,1,2,3> and run of pages <4,5,6> and creates run of pages <0,1,2,3,4,5,6>.
- Now the whole file is sorted.
- Main.cpp: Line 29 calls writeDiskFile() which writes DiskFile on cout.