Explaining the 2PMMS

- To calculate the block size, we proceed by initializing the memory to 2MB and integer size as 4 bytes.

  ```
  int memorySize = 2 * 1024;
  int integerSize = 4;
  int numberOfBlocks = n * integerSize / memorySize + ((n * integerSize / memorySize ==
  0) ? 0 : 1);
  int numberOfIntegers = memorySize/integerSize;
  ```

- numberOfIntegers indicate how many numbers will be stored in a 1 block.
- After creating this chunk of numbers we sort it and store them in a file.
- This process is continued until all the integers in the main input file are divided and sorted in their own files.
- This terminates the **phase 1** of the 2PMMS.
- The phase 2 of 2PMMS involves combining the separate sorted files.
- We create a combined file that contains the sorted numbers of both the first 2 sorted files.
- We subsequently create combined files for next iteration by combining the previously combined file and the next sorted file that was create from phase 1.
- The sorting of numbers in a combined files is done by maintaining 2 queues.
- The first queue holds the sorted numbers of the first file and the seconds holds the elements of the second file.
- We compare the front of both the queues as both these pointers will point to the least number in their respective files.
- As a result of the comparison, we get the least from both and add it to the combined sorted file.
- This continues till all the numbers from rest of the files have been added to the last combined file in the similar fashion.
- This concludes **phase 2** of 2PMMS.