Block Buffer:

Array of Structs:

struct Record{

str zip\_code,  
 place\_name,  
 state,  
 county,  
 latitude,  
 longitude;

};

Record Block[4]; //creates an array called Block, made up of 4 Record structs  
 //assuming 4 records per block

Private Methods:

1. Read\_Block\_from\_data( k1offset, k2offset, k3offset, k4offset );
   1. Read\_record( key\_offset ); X4 -> for each record
      1. Read\_zip\_code();
      2. Read\_place\_name();
      3. Read\_state();
      4. Read\_county();
      5. Read\_latitude();
      6. Read\_longitude();
2. Manipulation Methods for Block Data Stored in Memory
   1. Add\_record();
   2. Delete\_record();
   3. Update\_record();
   4. Display\_record();
   5. Compact\_record(); //will push all records to leftmost positions
3. Write\_Block\_to\_data( k1offset, k2offset, k3offset, k4offset);
   1. Write\_record(); X4
      1. Write\_zip\_code();
      2. Write\_place\_name();
      3. Write\_state();
      4. Write\_county();
      5. Write\_latitude();
      6. Write\_longitude();

User Functions:

1. User\_Delete\_record( primary key of record );
   1. Binary Search Index\_File for correct block
   2. Once the block is found take the RBN of the block
   3. Use offset of RBN to jump to the corresponding Block in the SS\_File
   4. Read k1offset, k2offset, k3offset, k4offset from SS\_File to get location of each record in the data.txt file.
   5. Read\_Block\_from\_data( k1offset, k2offset, k3offset, k4offset);
   6. Find the desired record from the block.
   7. Delete\_record(); -> in memory
      1. Save offset of deleted record.
   8. Compact\_record(); -> in memory
   9. Delete contents of the record from data.txt -> replace with spaces
   10. Write deleted record offset to deleted records file.
   11. Update the SS\_File with the updated block data -> replace with spaces
2. User\_Add\_record( record to add )