**kStacks**

**Q:** Create a data structure kStacks that represents k stacks. Implementation of kStacks should use only one array, i.e., k stacks should use the same array for storing elements. Following functions must be supported by kStacks.

push(int x, int sn) –> pushes x to stack number ‘sn’ where sn is from 0 to k-1

pop(int sn) –> pops an element from stack number ‘sn’ where sn is from 0 to k-1

**HINT:** Divide the array in slots of size n/k

**Algorithm:**

* Convert the strings to a Character array ( **Hint:** check toCharArray() )
* Create two arrays arr[ ] and index[ ].
* Divide the array arr[ ] into slots of n/k each with n being the size of the array and k being the number of stacks i.e. arr[0] to arr[(n/k)-1] will represent the first stack, arr[(n/k)] to arr[(2n/k)-1] will represent the second stack and so on..
* Store all the stack elements in array arr[ ].
* Store the indexes of the elements which are pushed to the stacks in array index [ ].

**Example:**

If the size of array arr[ ] is 50 and 5 stacks has to be implemented than array arr[ ] will break into size of 10 each, each sub-array representing one stack.

**Complexity:**

Time complexity: O(k) (Due to the for loop in constructor.)