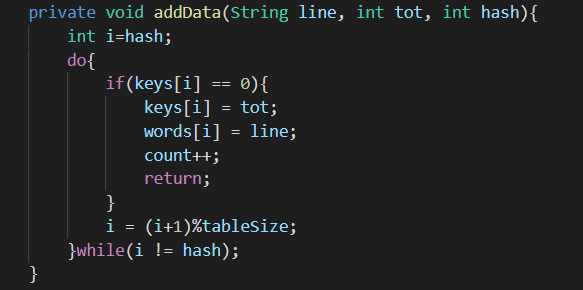
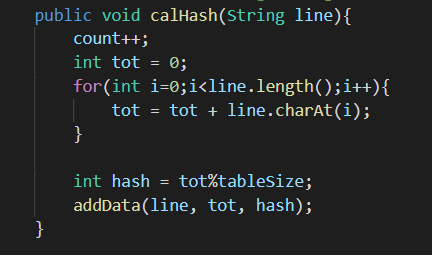
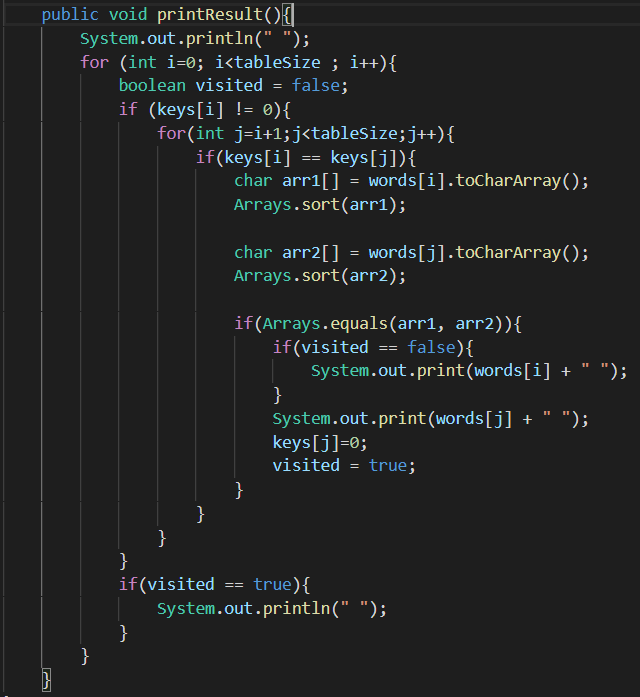
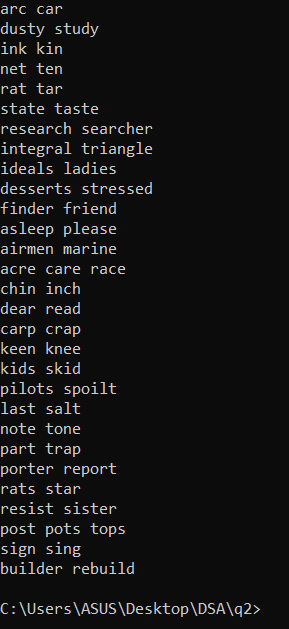
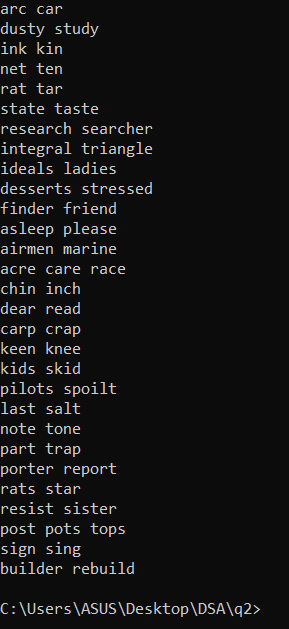
**Q2.** For the second one, we use java as the programming language.

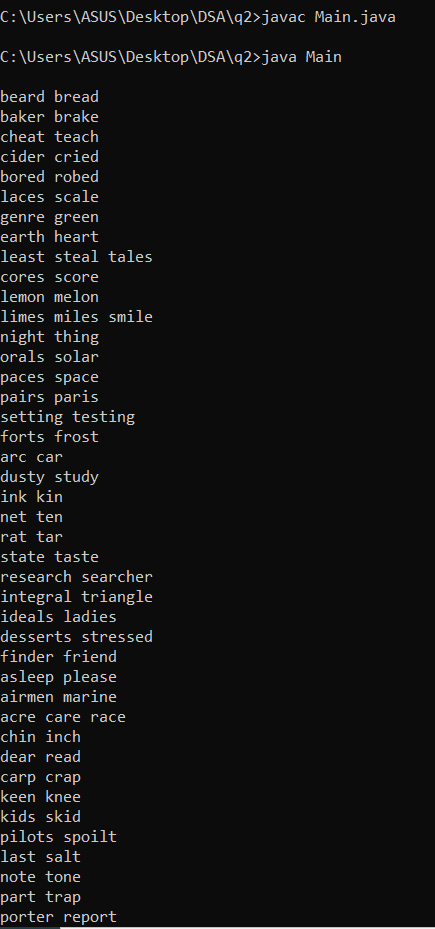
First the file is read line by line in the Main method and the calHash function in HashTable class is called after a line is read in Wordlist.txt file.

The calHash() method first converts the String values into integers and the total value of the characters are calculated. Then a hash key is generated for each word using the total value. Then the addData() method is called by passing the word, total and hash key for the respective word.The addData() method inserts the values to the hashtable. We have used open addressing and linear probing to avoid the collisions.

Here in the printResult() method, first we remove the keys with null values. Then another loop is carried out to check whether the given key is equal to another key. When such a place is found the first word is stored in arr1 and the other word is stored in arr2. Both arrays are sorted and check whether the characters are equal to each other. If so the words will be printed accordingly. In order to avoid the first letter being printed again, boolean value for visited variable is used. Only if the boolean value of visited is false the words will be printed, if true it will skip to the next line.



**Test cases :**



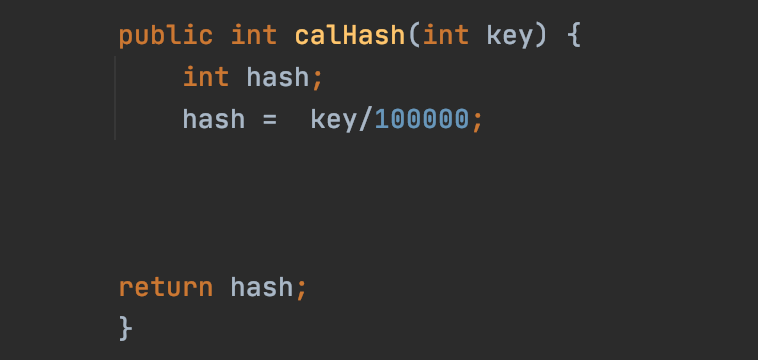
**Q3**. For Question no. 3 we used java as our programming language and chaining approach to make the hash table by using an array of linked lists.

We used ArrayList<Integer> arr = new ArrayList<Integer>(); to store data from the Pricelist.text document and to read the file we import java.io.File;. And we used “Scanner” to scan the data in the file and user inputs.

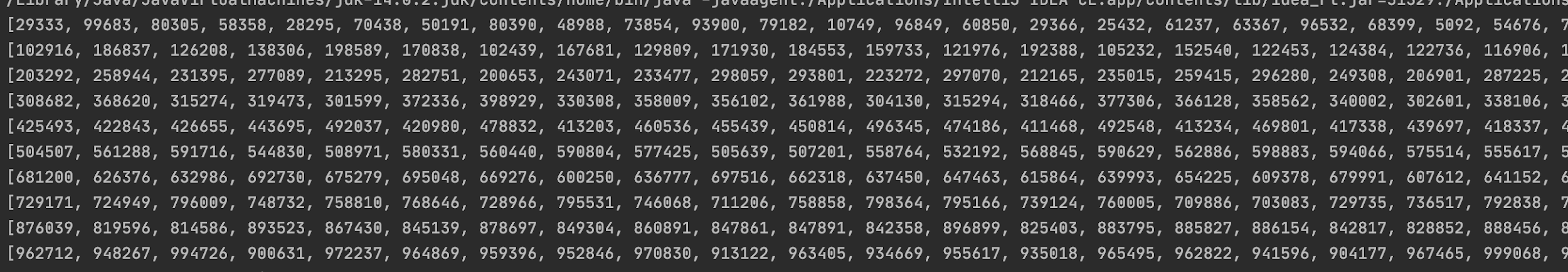
File file = new File("/Users/manthilabandara/Documents/manthila/Hashtable/q3/src/PriceList.txt”);

Scanner myReader = new Scanner(file);

After scanning data in the pricelist.txt and adding those data into the ArrayList Using a for loop we can calculate the hash key and add the data into the respective linked list. As there are 100000 data from 1 - 1000000 price range I select table size as 10 so I can divide the Prices by 100000 and get the hash Key from 0-9.

When adding data to the table first we get the data from the array and calculate its hash value and the hash value is equal to the position of the array. If the linked list in the position of the array is empty the value is added as the first value to the linked list. Else it will add to the end of the linked list. 

After printing the hash table it will look like this values from 1-100000 in the first line (0th Position) and 100000 - 200000 in the 2nd line (1st Position) and respectively.



After that we can take the user inputs to the program.

First i took the input as a string because user can input multiple inputs in a single line,

Then I split the string whenever there is a space and put it into an array(String[] str = val.trim().split("\\s+");). Then I parse those Strings into Integers. Then I sort the hash table as I think it will reduce the runtime when we find the values. Then I call the “findVal” method passing the user input and get the return value to the variable ans.

In the “findVal” method first we get the position by calling the “calHash” method as we don't need the values that are more than the user input as they always give a negative value.

So in the first for loop. We loop until the position of the input variable (“val”). And in the second for loop we go through the linked list, hash[i].size() means size of the linked list

We get values from each linked list and equal that data into the variable “w”( int w = hash[i].get(y);) 

and we subtract w by input variable val and get the remainder as “temp” (temp = val - w;

). If the remainder is negative we can continue the loop. else we can calculate the hash value of the temp and get the hash key of the temp value(x = calHash(temp);). Then we check whether the linked list in that key contains the temp value. If it contains the “temp” value then we make c =1 and break the loops and if there is no “temp” value c=0 and return the c value

And I put the C values into and array and display the result

These are Some test Cases