323.36 Project 2: Linked list implementation of Stack and Queue Java

Student: Jonathan Mathew
Project Due Date 09/15/2022

Algorithm Steps

Step 1: inFile→open input file using args [0]

bucketSize→args [1]
outFile1, outFile2→open output files using args [2] and args [3]

Step 2: createHashTable (...) // use constructor Step 3: informationProcessing (inFile, outFile2) Step 4: printHashTable (outFile1)
Step 5: close all files

Source Code

Main.java

```
import java.io.*;

import java.nio.file.Files;

public class Main {

    public static void main(String[] args) throws IOException {

        File inFile = new File(args[0]);

        int bucketSize = Integer.parseInt(args[1]);

        File out1 = new File(args[2]);

        File out2 = new File(args[3]);

        FileWriter outFile1 = new FileWriter(out1);

        FileWriter outFile2 = new FileWriter(out2);

        hashTable proj2 = new hashTable(bucketSize);

        proj2.informationProcessing(inFile, outFile2);

        proj2.printHashTable(outFile1);

        outFile1.close();

        outFile2.close();

    }

}
```

listNode.java

```
public class listNode {
    String data;
    listNode next;
    public listNode(String data){
        this.data = data;
    }
}
```

HashTable.java

```
import java.io.*;


public class hashTable {
    char op;
    String data;
    int bucketSize;
    listNode [] hashTable;
    public hashTable(int bucketSize){
        this.bucketSize = bucketSize;
        this.hashTable = new listNode[bucketSize];


        for(int i=0;i<bucketSize;i++){
            listNode node = new listNode("dummy");
            hashTable[i] = node;
        }
    }
    public int Doit(String data){
            long value = 1;
//           int value = 1;
            for(int i=0; i<data.length(); i++){
                char oneCh = data.charAt(i);
                value = value * 32 + (int) oneCh;
            }
//        System.out.println("value is:" + value);
            long spot = value % bucketSize;
            return (int)spot;
    }
    public listNode findSpot(int index, String data){
        listNode spot = hashTable[index];
        while(spot.next != null && spot.next.data.compareTo(data)<0)
            spot=spot.next;
        return spot;
    }

    public void hashInsert(int index, String data, FileWriter outFile2)
throws IOException {
        listNode spot = findSpot(index, data);
        outFile2.write("*** Inside hashInsert method. Performing
hashInsert \n");
```

```
        if(spot.next !=null && data.equals(spot.next.data)){
//            System.out.println("*** Warning, data is already in the
database!");
            outFile2.write("*** Warning, data is already in the database!
\n");
        }else{
            listNode newNode = new listNode(data);
            newNode.next = spot.next;
            spot.next = newNode;
            //output something from the file;
            outFile2.write("After hashInsert operation ... \n");
            printList(index, outFile2);
        }
    }
    public void hashDelete(int index, String data, FileWriter outFile2)
throws IOException {
        //output message here
        listNode spot = findSpot(index, data);
        outFile2.write("** Inside hashDelete method. Performing hashDelete
\n");
        if(spot.next == null || !spot.next.data.equals(data) ){
//            System.out.println("*** Warning: data is *not* in the
database!");
            outFile2.write("*** Warning: data is *not* in the database!
\n");
        }else{
            listNode temp = spot.next;
            spot.next =temp.next;
            temp.next =null;
            outFile2.write("After hashDelete operation ... \n");
            printList(index, outFile2);
        }
    }
    public void hashRetrieval(int index, String data, FileWriter outFile2)
throws IOException {
        //output file here
        listNode spot = findSpot(index, data);
        outFile2.write("** Inside hashRetrieval. Performing hashRetrieval
\n");
        if(spot.next == null || ! spot.next.data.equals(data)){
            //output file here
//              System.out.println("*** Warning, the record is *not* in the
database!");
            outFile2.write("*** Warning, the record is *not* in the
database! \n");
        }else{
            //outputfile here
//              System.out.println("Yes, the record is in the database!");
```

```
            outFile2.write("Yes, the record is in the database! \n");
        }
    }
    private void printList(int index, FileWriter outFile2) throws
IOException {
        listNode print = hashTable[index];
        StringBuilder pl=new StringBuilder();
        listNode temp = print;
        outFile2.write("HashTable[0]:  ");
        while(print.next !=null){

pl.append("(").append(print.data).append(",").append(print.next.data).append(")--->");
            temp=print;
            print=print.next;
        }
if(temp.next != null)
           temp = temp.next;
        pl.append("(").append(temp.data).append(",NULL)");
        outFile2.write(String.valueOf(pl.append("\n")));
//        System.out.println("NULL");
//        System.out.println("*************************");
    }
    public void printHashTable(FileWriter outFile1) throws IOException {
        for(int i=0; i<hashTable.length;i++){
            outFile1.write("HashTable["+ i + "]: ");

            printList(i,outFile1);

        }

    }


    public void informationProcessing(File inFile, FileWriter outFile2)
throws IOException {

        char op; //get from file

        String data;//get from file

        //print index here

//        printList(index, outfile);

        FileReader fr=new FileReader(inFile);   //reads the file

        BufferedReader br=new BufferedReader(fr);  //creates a buffering
character input stream

        //constructs a string buffer with no characters

        String line;

        outFile2.write("Enter informationProcessing method \n");

        while((line=br.readLine())!=null){

            outFile2.write("input is:   " + line+ "\n");

//            System.out.println("line data is:========   " + line);

            op = line.charAt(0);

            data = line.substring(1).trim();

            int index = Doit(data);

//            System.out.println("data is: "+ data);
//          System.out.println(index);
            outFile2.write("index for data is:  " + 0 + "\n");
            outFile2.write("linked list before insertion: \n");
            printList(index, outFile2);
            if(op == '+') {
                hashInsert(index, data, outFile2);
            }else if(op == '-'){
                hashDelete(index, data, outFile2);
            }else if(op == '?'){
                hashRetrieval(index, data, outFile2);
            }else{
                outFile2.write(op + " is an unrecognizable operation!"+
"\n");
            }
            outFile2.write("\n \n");
        }
    }
}
```

Output Bucket Size:7

Output File 1

```
HashTable[0]: (dummy,Cole)--->(Cole,David)--->(David,Jesse)--->(Jesse,Kevin)--->(Kevin,Ryan)--->(Ryan,Tsering)---
>(Tsering,NULL)

HashTable[1]: (dummy,Arben)--->(Arben,Benjamin)--->(Benjamin,Frederick)--->(Frederick,Hengtuo)--->(Hengtuo,Jia)---
>(Jia,Liwen)--->(Liwen,NULL)

HashTable[2]: (dummy,Aviraj)--->(Aviraj,Jurgen)--->(Jurgen,Seungwon)--->(Seungwon,Shaxzod)--->(Shaxzod,Yiqing)---
>(Yiqing,Zai)--->(Zai,NULL)

HashTable[3]: (dummy,Sean)--->(Sean,Siliang)--->(Siliang,Tania)--->(Tania,NULL)

HashTable[4]: (dummy,Alex)--->(Alex,Matthew)--->(Matthew,NULL)

HashTable[5]: (dummy,Diego)--->(Diego,Manana)--->(Manana,Mohammad)--->(Mohammad,Steven)--->(Steven,Tim)---
>(Tim,NULL)

HashTable[6]: (dummy,Daniel)--->(Daniel,Ivan)--->(Ivan,Jesrhythm)--->(Jesrhythm,Tshetrim)--->(Tshetrim,NULL)
```

Output File 2

```
Enter informationProcessing method

input is:   +                  Mohammad

index for data is:  5

linked list before insertion:

HashTable[5]: (dummy,NULL)

*** Inside hashInsert method. Performing hashInsert

After hashInsert operation ...

HashTable[5]: (dummy,Mohammad)--->(Mohammad,NULL)




input is:   +                  Diego

index for data is:  5

linked list before insertion:

HashTable[5]: (dummy,Mohammad)--->(Mohammad,NULL)

*** Inside hashInsert method. Performing hashInsert

After hashInsert operation ...

HashTable[5]: (dummy,Diego)--->(Diego,Mohammad)--->(Mohammad,NULL)
```

```
input is:   +                  Daniel

index for data is:  6

linked list before insertion:

HashTable[6]: (dummy,NULL)

*** Inside hashInsert method. Performing hashInsert

After hashInsert operation ...

HashTable[6]: (dummy,Daniel)--->(Daniel,NULL)



input is:   +                  Alex

index for data is:  4

linked list before insertion:

HashTable[4]: (dummy,NULL)

*** Inside hashInsert method. Performing hashInsert

After hashInsert operation ...

HashTable[4]: (dummy,Alex)--->(Alex,NULL)



input is:   +                  AlAmin

index for data is:  4

linked list before insertion:

HashTable[4]: (dummy,Alex)--->(Alex,NULL)

*** Inside hashInsert method. Performing hashInsert

After hashInsert operation ...

HashTable[4]: (dummy,AlAmin)--->(AlAmin,Alex)--->(Alex,NULL)
```

```
input is:   ?AlAmin

index for data is:  4

linked list before insertion:

HashTable[4]: (dummy,AlAmin)--->(AlAmin,Alex)--->(Alex,NULL)

** Inside hashRetrieval. Performing hashRetrieval

Yes, the record is in the database!



input is:   - AlAmin

index for data is:  4

linked list before insertion:

HashTable[4]: (dummy,AlAmin)--->(AlAmin,Alex)--->(Alex,NULL)

** Inside hashDelete method. Performing hashDelete

After hashDelete operation ...

HashTable[4]: (dummy,Alex)--->(Alex,NULL)



input is:   ?AlAmin

index for data is:  4

linked list before insertion:

HashTable[4]: (dummy,Alex)--->(Alex,NULL)

** Inside hashRetrieval. Performing hashRetrieval

*** Warning, the record is *not* in the database!



input is:   +                 Cole

index for data is:  0

linked list before insertion:
```

```
HashTable[0]: (dummy,NULL)

*** Inside hashInsert method. Performing hashInsert

After hashInsert operation ...

HashTable[0]: (dummy,Cole)--->(Cole,NULL)




input is:   +               Jia

index for data is:  1

linked list before insertion:

HashTable[1]: (dummy,NULL)

*** Inside hashInsert method. Performing hashInsert

After hashInsert operation ...

HashTable[1]: (dummy,Jia)--->(Jia,NULL)




input is:   +               Liwen

index for data is:  1

linked list before insertion:

HashTable[1]: (dummy,Jia)--->(Jia,NULL)

*** Inside hashInsert method. Performing hashInsert

After hashInsert operation ...

HashTable[1]: (dummy,Jia)--->(Jia,Liwen)--->(Liwen,NULL)




input is:   +               Jesse

index for data is:  0

linked list before insertion:

HashTable[0]: (dummy,Cole)--->(Cole,NULL)
```

```
*** Inside hashInsert method. Performing hashInsert

After hashInsert operation ...

HashTable[0]: (dummy,Cole)--->(Cole,Jesse)--->(Jesse,NULL)




input is:   ?Jesse

index for data is:  0

linked list before insertion:

HashTable[0]: (dummy,Cole)--->(Cole,Jesse)--->(Jesse,NULL)

** Inside hashRetrieval. Performing hashRetrieval

Yes, the record is in the database!




input is:   - Jesse

index for data is:  0

linked list before insertion:

HashTable[0]: (dummy,Cole)--->(Cole,Jesse)--->(Jesse,NULL)

** Inside hashDelete method. Performing hashDelete

After hashDelete operation ...

HashTable[0]: (dummy,Cole)--->(Cole,NULL)




input is:   +                  Jesse

index for data is:  0

linked list before insertion:

HashTable[0]: (dummy,Cole)--->(Cole,NULL)

*** Inside hashInsert method. Performing hashInsert

After hashInsert operation ...
```

```
HashTable[0]: (dummy,Cole)--->(Cole,Jesse)--->(Jesse,NULL)



input is:   +                  Ning

index for data is:  2

linked list before insertion:

HashTable[2]: (dummy,NULL)

*** Inside hashInsert method. Performing hashInsert

After hashInsert operation ...

HashTable[2]: (dummy,Ning)--->(Ning,NULL)



input is:   ?Ning

index for data is:  2

linked list before insertion:

HashTable[2]: (dummy,Ning)--->(Ning,NULL)

** Inside hashRetrieval. Performing hashRetrieval

Yes, the record is in the database!



input is:   +                  Asadbek

index for data is:  3

linked list before insertion:

HashTable[3]: (dummy,NULL)

*** Inside hashInsert method. Performing hashInsert

After hashInsert operation ...

HashTable[3]: (dummy,Asadbek)--->(Asadbek,NULL)
```

```
input is:   - Asadbek
index for data is:  3
linked list before insertion:
HashTable[3]: (dummy,Asadbek)--->(Asadbek,NULL)
** Inside hashDelete method. Performing hashDelete
After hashDelete operation ...
HashTable[3]: (dummy,NULL)


input is:   - Asadbek
index for data is:  3
linked list before insertion:
HashTable[3]: (dummy,NULL)
** Inside hashDelete method. Performing hashDelete
*** Warning: data is *not* in the database!


input is:   *               Asadbek
index for data is:  3
linked list before insertion:
HashTable[3]: (dummy,NULL)
* is an unrecognizable operation!


input is:   +               Jesrhythm
index for data is:  6
linked list before insertion:
```

```
HashTable[6]: (dummy,Daniel)--->(Daniel,NULL)

*** Inside hashInsert method. Performing hashInsert

After hashInsert operation ...

HashTable[6]: (dummy,Daniel)--->(Daniel,Jesrhythm)--->(Jesrhythm,NULL)




input is:   - Ning

index for data is:  2

linked list before insertion:

HashTable[2]: (dummy,Ning)--->(Ning,NULL)

** Inside hashDelete method. Performing hashDelete

After hashDelete operation ...

HashTable[2]: (dummy,NULL)




input is:   +              Ryan

index for data is:  0

linked list before insertion:

HashTable[0]: (dummy,Cole)--->(Cole,Jesse)--->(Jesse,NULL)

*** Inside hashInsert method. Performing hashInsert

After hashInsert operation ...

HashTable[0]: (dummy,Cole)--->(Cole,Jesse)--->(Jesse,Ryan)--->(Ryan,NULL)




input is:   +              Asadbek

index for data is:  3

linked list before insertion:

HashTable[3]: (dummy,NULL)
```

```
*** Inside hashInsert method. Performing hashInsert

After hashInsert operation ...

HashTable[3]: (dummy,Asadbek)--->(Asadbek,NULL)




input is:   +               Tsering

index for data is:  0

linked list before insertion:

HashTable[0]: (dummy,Cole)--->(Cole,Jesse)--->(Jesse,Ryan)--->(Ryan,NULL)

*** Inside hashInsert method. Performing hashInsert

After hashInsert operation ...

HashTable[0]: (dummy,Cole)--->(Cole,Jesse)--->(Jesse,Ryan)--->(Ryan,Tsering)--->(Tsering,NULL)




input is:   +               Tshetrim

index for data is:  6

linked list before insertion:

HashTable[6]: (dummy,Daniel)--->(Daniel,Jesrhythm)--->(Jesrhythm,NULL)

*** Inside hashInsert method. Performing hashInsert

After hashInsert operation ...

HashTable[6]: (dummy,Daniel)--->(Daniel,Jesrhythm)--->(Jesrhythm,Tshetrim)--->(Tshetrim,NULL)




input is:   +               Yiqing

index for data is:  2

linked list before insertion:

HashTable[2]: (dummy,NULL)

*** Inside hashInsert method. Performing hashInsert
```

```
After hashInsert operation ...

HashTable[2]: (dummy,Yiqing)--->(Yiqing,NULL)




input is:   - Asadbek

index for data is:  3

linked list before insertion:

HashTable[3]: (dummy,Asadbek)--->(Asadbek,NULL)

** Inside hashDelete method. Performing hashDelete

After hashDelete operation ...

HashTable[3]: (dummy,NULL)




input is:   +                 Manana

index for data is:  5

linked list before insertion:

HashTable[5]: (dummy,Diego)--->(Diego,Mohammad)--->(Mohammad,NULL)

*** Inside hashInsert method. Performing hashInsert

After hashInsert operation ...

HashTable[5]: (dummy,Diego)--->(Diego,Manana)--->(Manana,Mohammad)--->(Mohammad,NULL)




input is:   +                 Ivan

index for data is:  6

linked list before insertion:

HashTable[6]: (dummy,Daniel)--->(Daniel,Jesrhythm)--->(Jesrhythm,Tshetrim)--->(Tshetrim,NULL)

*** Inside hashInsert method. Performing hashInsert

After hashInsert operation ...
```

```
HashTable[6]: (dummy,Daniel)--->(Daniel,Ivan)--->(Ivan,Jesrhythm)--->(Jesrhythm,Tshetrim)--->(Tshetrim,NULL)




input is:   +           Tania

index for data is:  3

linked list before insertion:

HashTable[3]: (dummy,NULL)

*** Inside hashInsert method. Performing hashInsert

After hashInsert operation ...

HashTable[3]: (dummy,Tania)--->(Tania,NULL)




input is:   %           Aniss

index for data is:  2

linked list before insertion:

HashTable[2]: (dummy,Yiqing)--->(Yiqing,NULL)

% is an unrecognizable operation!




input is:   ?Erik

index for data is:  2

linked list before insertion:

HashTable[2]: (dummy,Yiqing)--->(Yiqing,NULL)

** Inside hashRetrieval. Performing hashRetrieval

*** Warning, the record is *not* in the database!




input is:   +           Erik
```

```
index for data is:  2

linked list before insertion:

HashTable[2]: (dummy,Yiqing)--->(Yiqing,NULL)

*** Inside hashInsert method. Performing hashInsert

After hashInsert operation ...

HashTable[2]: (dummy,Erik)--->(Erik,Yiqing)--->(Yiqing,NULL)




input is:   - Erik

index for data is:  2

linked list before insertion:

HashTable[2]: (dummy,Erik)--->(Erik,Yiqing)--->(Yiqing,NULL)

** Inside hashDelete method. Performing hashDelete

After hashDelete operation ...

HashTable[2]: (dummy,Yiqing)--->(Yiqing,NULL)




input is:   ?Erik

index for data is:  2

linked list before insertion:

HashTable[2]: (dummy,Yiqing)--->(Yiqing,NULL)

** Inside hashRetrieval. Performing hashRetrieval

*** Warning, the record is *not* in the database!




input is:   +                 Kevin

index for data is:  0

linked list before insertion:
```

```
HashTable[0]: (dummy,Cole)--->(Cole,Jesse)--->(Jesse,Ryan)--->(Ryan,Tsering)--->(Tsering,NULL)

*** Inside hashInsert method. Performing hashInsert

After hashInsert operation ...

HashTable[0]: (dummy,Cole)--->(Cole,Jesse)--->(Jesse,Kevin)--->(Kevin,Ryan)--->(Ryan,Tsering)--->(Tsering,NULL)




input is:   +                  Arben

index for data is:  1

linked list before insertion:

HashTable[1]: (dummy,Jia)--->(Jia,Liwen)--->(Liwen,NULL)

*** Inside hashInsert method. Performing hashInsert

After hashInsert operation ...

HashTable[1]: (dummy,Arben)--->(Arben,Jia)--->(Jia,Liwen)--->(Liwen,NULL)




input is:   +                  David

index for data is:  0

linked list before insertion:

HashTable[0]: (dummy,Cole)--->(Cole,Jesse)--->(Jesse,Kevin)--->(Kevin,Ryan)--->(Ryan,Tsering)--->(Tsering,NULL)

*** Inside hashInsert method. Performing hashInsert

After hashInsert operation ...

HashTable[0]: (dummy,Cole)--->(Cole,David)--->(David,Jesse)--->(Jesse,Kevin)--->(Kevin,Ryan)--->(Ryan,Tsering)--->(Tsering,NULL)




input is:   +                  Benjamin

index for data is:  1

linked list before insertion:

HashTable[1]: (dummy,Arben)--->(Arben,Jia)--->(Jia,Liwen)--->(Liwen,NULL)
```

```
*** Inside hashInsert method. Performing hashInsert

After hashInsert operation ...

HashTable[1]: (dummy,Arben)--->(Arben,Benjamin)--->(Benjamin,Jia)--->(Jia,Liwen)--->(Liwen,NULL)




input is:   - Fahim

index for data is:  3

linked list before insertion:

HashTable[3]: (dummy,Tania)--->(Tania,NULL)

** Inside hashDelete method. Performing hashDelete

*** Warning: data is *not* in the database!




input is:   +                Hengtuo

index for data is:  1

linked list before insertion:

HashTable[1]: (dummy,Arben)--->(Arben,Benjamin)--->(Benjamin,Jia)--->(Jia,Liwen)--->(Liwen,NULL)

*** Inside hashInsert method. Performing hashInsert

After hashInsert operation ...

HashTable[1]: (dummy,Arben)--->(Arben,Benjamin)--->(Benjamin,Hengtuo)--->(Hengtuo,Jia)--->(Jia,Liwen)--->(Liwen,NULL)




input is:   +                Siliang

index for data is:  3

linked list before insertion:

HashTable[3]: (dummy,Tania)--->(Tania,NULL)

*** Inside hashInsert method. Performing hashInsert

After hashInsert operation ...
```

```
HashTable[3]: (dummy,Siliang)--->(Siliang,Tania)--->(Tania,NULL)




input is:   +                 Seungwon

index for data is:  2

linked list before insertion:

HashTable[2]: (dummy,Yiqing)--->(Yiqing,NULL)

*** Inside hashInsert method. Performing hashInsert

After hashInsert operation ...

HashTable[2]: (dummy,Seungwon)--->(Seungwon,Yiqing)--->(Yiqing,NULL)




input is:   ?Yahav

index for data is:  0

linked list before insertion:

HashTable[0]: (dummy,Cole)--->(Cole,David)--->(David,Jesse)--->(Jesse,Kevin)--->(Kevin,Ryan)--->(Ryan,Tsering)--->(Tsering,NULL)

** Inside hashRetrieval. Performing hashRetrieval

*** Warning, the record is *not* in the database!




input is:   +                 Jurgen

index for data is:  2

linked list before insertion:

HashTable[2]: (dummy,Seungwon)--->(Seungwon,Yiqing)--->(Yiqing,NULL)

*** Inside hashInsert method. Performing hashInsert

After hashInsert operation ...

HashTable[2]: (dummy,Jurgen)--->(Jurgen,Seungwon)--->(Seungwon,Yiqing)--->(Yiqing,NULL)
```

```
input is:   -Gildian

index for data is:  6

linked list before insertion:

HashTable[6]: (dummy,Daniel)--->(Daniel,Ivan)--->(Ivan,Jesrhythm)--->(Jesrhythm,Tshetrim)--->(Tshetrim,NULL)

** Inside hashDelete method. Performing hashDelete

*** Warning: data is *not* in the database!




input is:   ?Seong

index for data is:  3

linked list before insertion:

HashTable[3]: (dummy,Siliang)--->(Siliang,Tania)--->(Tania,NULL)

** Inside hashRetrieval. Performing hashRetrieval

*** Warning, the record is *not* in the database!




input is:   #           Auyon

index for data is:  6

linked list before insertion:

HashTable[6]: (dummy,Daniel)--->(Daniel,Ivan)--->(Ivan,Jesrhythm)--->(Jesrhythm,Tshetrim)--->(Tshetrim,NULL)

# is an unrecognizable operation!




input is:   +           Frederick

index for data is:  1

linked list before insertion:

HashTable[1]: (dummy,Arben)--->(Arben,Benjamin)--->(Benjamin,Hengtuo)--->(Hengtuo,Jia)--->(Jia,Liwen)--->(Liwen,NULL)
```

```
*** Inside hashInsert method. Performing hashInsert

After hashInsert operation ...

HashTable[1]: (dummy,Arben)--->(Arben,Benjamin)--->(Benjamin,Frederick)--->(Frederick,Hengtuo)--->(Hengtuo,Jia)--->(Jia,Liwen)---
>(Liwen,NULL)




input is:   +                 Aviraj

index for data is:  2

linked list before insertion:

HashTable[2]: (dummy,Jurgen)--->(Jurgen,Seungwon)--->(Seungwon,Yiqing)--->(Yiqing,NULL)

*** Inside hashInsert method. Performing hashInsert

After hashInsert operation ...

HashTable[2]: (dummy,Aviraj)--->(Aviraj,Jurgen)--->(Jurgen,Seungwon)--->(Seungwon,Yiqing)--->(Yiqing,NULL)




input is:   +                 Shaxzod

index for data is:  2

linked list before insertion:

HashTable[2]: (dummy,Aviraj)--->(Aviraj,Jurgen)--->(Jurgen,Seungwon)--->(Seungwon,Yiqing)--->(Yiqing,NULL)

*** Inside hashInsert method. Performing hashInsert

After hashInsert operation ...

HashTable[2]: (dummy,Aviraj)--->(Aviraj,Jurgen)--->(Jurgen,Seungwon)--->(Seungwon,Shaxzod)--->(Shaxzod,Yiqing)--->(Yiqing,NULL)




input is:   - Imran

index for data is:  2

linked list before insertion:

HashTable[2]: (dummy,Aviraj)--->(Aviraj,Jurgen)--->(Jurgen,Seungwon)--->(Seungwon,Shaxzod)--->(Shaxzod,Yiqing)--->(Yiqing,NULL)

** Inside hashDelete method. Performing hashDelete
```

```
*** Warning: data is *not* in the database!




input is:   +                  Zai

index for data is:  2

linked list before insertion:

HashTable[2]: (dummy,Aviraj)--->(Aviraj,Jurgen)--->(Jurgen,Seungwon)--->(Seungwon,Shaxzod)--->(Shaxzod,Yiqing)--->(Yiqing,NULL)

*** Inside hashInsert method. Performing hashInsert

After hashInsert operation ...

HashTable[2]: (dummy,Aviraj)--->(Aviraj,Jurgen)--->(Jurgen,Seungwon)--->(Seungwon,Shaxzod)--->(Shaxzod,Yiqing)--->(Yiqing,Zai)--->(Zai,NULL)




input is:   ?Jonathan

index for data is:  5

linked list before insertion:

HashTable[5]: (dummy,Diego)--->(Diego,Manana)--->(Manana,Mohammad)--->(Mohammad,NULL)

** Inside hashRetrieval. Performing hashRetrieval

*** Warning, the record is *not* in the database!




input is:   +                  Sean

index for data is:  3

linked list before insertion:

HashTable[3]: (dummy,Siliang)--->(Siliang,Tania)--->(Tania,NULL)

*** Inside hashInsert method. Performing hashInsert

After hashInsert operation ...

HashTable[3]: (dummy,Sean)--->(Sean,Siliang)--->(Siliang,Tania)--->(Tania,NULL)
```

```
input is:   - Brandon

index for data is:  3

linked list before insertion:

HashTable[3]: (dummy,Sean)--->(Sean,Siliang)--->(Siliang,Tania)--->(Tania,NULL)

** Inside hashDelete method. Performing hashDelete

*** Warning: data is *not* in the database!




input is:   ?Juan

index for data is:  5

linked list before insertion:

HashTable[5]: (dummy,Diego)--->(Diego,Manana)--->(Manana,Mohammad)--->(Mohammad,NULL)

** Inside hashRetrieval. Performing hashRetrieval

*** Warning, the record is *not* in the database!




input is:   +                 Steven

index for data is:  5

linked list before insertion:

HashTable[5]: (dummy,Diego)--->(Diego,Manana)--->(Manana,Mohammad)--->(Mohammad,NULL)

*** Inside hashInsert method. Performing hashInsert

After hashInsert operation ...

HashTable[5]: (dummy,Diego)--->(Diego,Manana)--->(Manana,Mohammad)--->(Mohammad,Steven)--->(Steven,NULL)




input is:   +                 Tim

index for data is:  5
```

```
linked list before insertion:

HashTable[5]: (dummy,Diego)--->(Diego,Manana)--->(Manana,Mohammad)--->(Mohammad,Steven)--->(Steven,NULL)

*** Inside hashInsert method. Performing hashInsert

After hashInsert operation ...

HashTable[5]: (dummy,Diego)--->(Diego,Manana)--->(Manana,Mohammad)--->(Mohammad,Steven)--->(Steven,Tim)--->(Tim,NULL)




input is:   +                 Matthew

index for data is:  4

linked list before insertion:

HashTable[4]: (dummy,Alex)--->(Alex,NULL)

*** Inside hashInsert method. Performing hashInsert

After hashInsert operation ...

HashTable[4]: (dummy,Alex)--->(Alex,Matthew)--->(Matthew,NULL)




input is:   +                 Natasha

index for data is:  5

linked list before insertion:

HashTable[5]: (dummy,Diego)--->(Diego,Manana)--->(Manana,Mohammad)--->(Mohammad,Steven)--->(Steven,Tim)--->(Tim,NULL)

*** Inside hashInsert method. Performing hashInsert

After hashInsert operation ...

HashTable[5]: (dummy,Diego)--->(Diego,Manana)--->(Manana,Mohammad)--->(Mohammad,Natasha)--->(Natasha,Steven)--->(Steven,Tim)---
>(Tim,NULL)




input is:   - Natasha

index for data is:  5

linked list before insertion:
```

```
HashTable[5]: (dummy,Diego)--->(Diego,Manana)--->(Manana,Mohammad)--->(Mohammad,Natasha)--->(Natasha,Steven)--->(Steven,Tim)--->(Tim,NULL)

** Inside hashDelete method. Performing hashDelete

After hashDelete operation ...

HashTable[5]: (dummy,Diego)--->(Diego,Manana)--->(Manana,Mohammad)--->(Mohammad,Steven)--->(Steven,Tim)--->(Tim,NULL)




input is:   ?Natasha

index for data is:  5

linked list before insertion:

HashTable[5]: (dummy,Diego)--->(Diego,Manana)--->(Manana,Mohammad)--->(Mohammad,Steven)--->(Steven,Tim)--->(Tim,NULL)

** Inside hashRetrieval. Performing hashRetrieval

*** Warning, the record is *not* in the database!
```

Output Bucket Size:13

Output File 1

```
HashTable[0]: (dummy,Liwen)--->(Liwen,Matthew)--->(Matthew,Siliang)--->(Siliang,Steven)--->(Steven,NULL)

HashTable[1]: (dummy,Tim)--->(Tim,NULL)

HashTable[2]: (dummy,NULL)

HashTable[3]: (dummy,NULL)

HashTable[4]: (dummy,Frederick)--->(Frederick,Tsering)--->(Tsering,Yiqing)--->(Yiqing,NULL)

HashTable[5]: (dummy,NULL)

HashTable[6]: (dummy,Jia)--->(Jia,Manana)--->(Manana,Ryan)--->(Ryan,Tshetrim)--->(Tshetrim,NULL)
```

```
HashTable[7]: (dummy,Tania)--->(Tania,NULL)

HashTable[8]: (dummy,Alex)--->(Alex,Diego)--->(Diego,Ivan)--->(Ivan,Jesrhythm)--->(Jesrhythm,NULL)

HashTable[9]: (dummy,Mohammad)--->(Mohammad,Sean)--->(Sean,Zai)--->(Zai,NULL)

HashTable[10]: (dummy,Arben)--->(Arben,Benjamin)--->(Benjamin,Jurgen)--->(Jurgen,NULL)

HashTable[11]: (dummy,Aviraj)--->(Aviraj,David)--->(David,Hengtuo)--->(Hengtuo,Jesse)--->(Jesse,NULL)

HashTable[12]: (dummy,Cole)--->(Cole,Daniel)--->(Daniel,Kevin)--->(Kevin,Seungwon)--->(Seungwon,Shaxzod)--->(Shaxzod,NULL)
```

Output File 2

```
Enter informationProcessing method

input is:   +       Mohammad

index for data is:  9

linked list before insertion:

HashTable[9]: (dummy,NULL)

*** Inside hashInsert method. Performing hashInsert

After hashInsert operation ...

HashTable[9]: (dummy,Mohammad)--->(Mohammad,NULL)




input is:   +       Diego

index for data is:  8

linked list before insertion:

HashTable[8]: (dummy,NULL)

*** Inside hashInsert method. Performing hashInsert
```

```
After hashInsert operation ...

HashTable[8]: (dummy,Diego)--->(Diego,NULL)




input is:   +       Daniel

index for data is:  12

linked list before insertion:

HashTable[12]: (dummy,NULL)

*** Inside hashInsert method. Performing hashInsert

After hashInsert operation ...

HashTable[12]: (dummy,Daniel)--->(Daniel,NULL)




input is:   +       Alex

index for data is:  8

linked list before insertion:

HashTable[8]: (dummy,Diego)--->(Diego,NULL)

*** Inside hashInsert method. Performing hashInsert

After hashInsert operation ...

HashTable[8]: (dummy,Alex)--->(Alex,Diego)--->(Diego,NULL)




input is:   +       AlAmin

index for data is:  6
```

```
linked list before insertion:
HashTable[6]: (dummy,NULL)
*** Inside hashInsert method. Performing hashInsert
After hashInsert operation ...
HashTable[6]: (dummy,AlAmin)--->(AlAmin,NULL)



input is:   ?         AlAmin
index for data is:  6
linked list before insertion:
HashTable[6]: (dummy,AlAmin)--->(AlAmin,NULL)
** Inside hashRetrieval. Performing hashRetrieval
Yes, the record is in the database!



input is:   -         AlAmin
index for data is:  6
linked list before insertion:
HashTable[6]: (dummy,AlAmin)--->(AlAmin,NULL)
** Inside hashDelete method. Performing hashDelete
After hashDelete operation ...
HashTable[6]: (dummy,NULL)
```

```
input is:   ?        AlAmin
index for data is:  6
linked list before insertion:
HashTable[6]: (dummy,NULL)
** Inside hashRetrieval. Performing hashRetrieval
*** Warning, the record is *not* in the database!



input is:   +        Cole
index for data is:  12
linked list before insertion:
HashTable[12]: (dummy,Daniel)--->(Daniel,NULL)
*** Inside hashInsert method. Performing hashInsert
After hashInsert operation ...
HashTable[12]: (dummy,Cole)--->(Cole,Daniel)--->(Daniel,NULL)



input is:   +        Jia
index for data is:  6
linked list before insertion:
HashTable[6]: (dummy,NULL)
*** Inside hashInsert method. Performing hashInsert
After hashInsert operation ...
HashTable[6]: (dummy,Jia)--->(Jia,NULL)
```

```
input is:   +        Liwen

index for data is:  0

linked list before insertion:

HashTable[0]: (dummy,NULL)

*** Inside hashInsert method. Performing hashInsert

After hashInsert operation ...

HashTable[0]: (dummy,Liwen)--->(Liwen,NULL)




input is:   +        Jesse

index for data is:  11

linked list before insertion:

HashTable[11]: (dummy,NULL)

*** Inside hashInsert method. Performing hashInsert

After hashInsert operation ...

HashTable[11]: (dummy,Jesse)--->(Jesse,NULL)




input is:   ?        Jesse

index for data is:  11

linked list before insertion:

HashTable[11]: (dummy,Jesse)--->(Jesse,NULL)
```

```
** Inside hashRetrieval. Performing hashRetrieval

Yes, the record is in the database!




input is:   -        Jesse

index for data is:  11

linked list before insertion:

HashTable[11]: (dummy,Jesse)--->(Jesse,NULL)

** Inside hashDelete method. Performing hashDelete

After hashDelete operation ...

HashTable[11]: (dummy,NULL)




input is:   +        Jesse

index for data is:  11

linked list before insertion:

HashTable[11]: (dummy,NULL)

*** Inside hashInsert method. Performing hashInsert

After hashInsert operation ...

HashTable[11]: (dummy,Jesse)--->(Jesse,NULL)




input is:   +        Ning

index for data is:  2
```

```
linked list before insertion:

HashTable[2]: (dummy,NULL)

*** Inside hashInsert method. Performing hashInsert

After hashInsert operation ...

HashTable[2]: (dummy,Ning)--->(Ning,NULL)



input is:   ?        Ning

index for data is:  2

linked list before insertion:

HashTable[2]: (dummy,Ning)--->(Ning,NULL)

** Inside hashRetrieval. Performing hashRetrieval

Yes, the record is in the database!



input is:   +        Asadbek

index for data is:  2

linked list before insertion:

HashTable[2]: (dummy,Ning)--->(Ning,NULL)

*** Inside hashInsert method. Performing hashInsert

After hashInsert operation ...

HashTable[2]: (dummy,Asadbek)--->(Asadbek,Ning)--->(Ning,NULL)
```

```
input is:   -         Asadbek
index for data is:  2
linked list before insertion:
HashTable[2]: (dummy,Asadbek)--->(Asadbek,Ning)--->(Ning,NULL)
** Inside hashDelete method. Performing hashDelete
After hashDelete operation ...
HashTable[2]: (dummy,Ning)--->(Ning,NULL)


input is:   -         Asadbek
index for data is:  2
linked list before insertion:
HashTable[2]: (dummy,Ning)--->(Ning,NULL)
** Inside hashDelete method. Performing hashDelete
*** Warning: data is *not* in the database!


input is:   *         Asadbek
index for data is:  2
linked list before insertion:
HashTable[2]: (dummy,Ning)--->(Ning,NULL)
* is an unrecognizable operation!
```

```
input is:   +        Jesrhythm
index for data is:  8
linked list before insertion:
HashTable[8]: (dummy,Alex)--->(Alex,Diego)--->(Diego,NULL)
*** Inside hashInsert method. Performing hashInsert
After hashInsert operation ...
HashTable[8]: (dummy,Alex)--->(Alex,Diego)--->(Diego,Jesrhythm)--->(Jesrhythm,NULL)



input is:   -        Ning
index for data is:  2
linked list before insertion:
HashTable[2]: (dummy,Ning)--->(Ning,NULL)
** Inside hashDelete method. Performing hashDelete
After hashDelete operation ...
HashTable[2]: (dummy,NULL)



input is:   +        Ryan
index for data is:  6
linked list before insertion:
HashTable[6]: (dummy,Jia)--->(Jia,NULL)
*** Inside hashInsert method. Performing hashInsert
After hashInsert operation ...
```

```
HashTable[6]: (dummy,Jia)--->(Jia,Ryan)--->(Ryan,NULL)



input is:   +       Asadbek

index for data is:  2

linked list before insertion:

HashTable[2]: (dummy,NULL)

*** Inside hashInsert method. Performing hashInsert

After hashInsert operation ...

HashTable[2]: (dummy,Asadbek)--->(Asadbek,NULL)



input is:   +       Tsering

index for data is:  4

linked list before insertion:

HashTable[4]: (dummy,NULL)

*** Inside hashInsert method. Performing hashInsert

After hashInsert operation ...

HashTable[4]: (dummy,Tsering)--->(Tsering,NULL)



input is:   +       Tshetrim

index for data is:  6

linked list before insertion:
```

```
HashTable[6]: (dummy,Jia)--->(Jia,Ryan)--->(Ryan,NULL)

*** Inside hashInsert method. Performing hashInsert

After hashInsert operation ...

HashTable[6]: (dummy,Jia)--->(Jia,Ryan)--->(Ryan,Tshetrim)--->(Tshetrim,NULL)




input is:   +        Yiqing

index for data is:  4

linked list before insertion:

HashTable[4]: (dummy,Tsering)--->(Tsering,NULL)

*** Inside hashInsert method. Performing hashInsert

After hashInsert operation ...

HashTable[4]: (dummy,Tsering)--->(Tsering,Yiqing)--->(Yiqing,NULL)




input is:   -        Asadbek

index for data is:  2

linked list before insertion:

HashTable[2]: (dummy,Asadbek)--->(Asadbek,NULL)

** Inside hashDelete method. Performing hashDelete

After hashDelete operation ...

HashTable[2]: (dummy,NULL)
```

```
input is:   +        Manana

index for data is:  6

linked list before insertion:

HashTable[6]: (dummy,Jia)--->(Jia,Ryan)--->(Ryan,Tshetrim)--->(Tshetrim,NULL)

*** Inside hashInsert method. Performing hashInsert

After hashInsert operation ...

HashTable[6]: (dummy,Jia)--->(Jia,Manana)--->(Manana,Ryan)--->(Ryan,Tshetrim)---
>(Tshetrim,NULL)




input is:   +        Ivan

index for data is:  8

linked list before insertion:

HashTable[8]: (dummy,Alex)--->(Alex,Diego)--->(Diego,Jesrhythm)--->(Jesrhythm,NULL)

*** Inside hashInsert method. Performing hashInsert

After hashInsert operation ...

HashTable[8]: (dummy,Alex)--->(Alex,Diego)--->(Diego,Ivan)--->(Ivan,Jesrhythm)---
>(Jesrhythm,NULL)




input is:   +        Tania

index for data is:  7

linked list before insertion:

HashTable[7]: (dummy,NULL)

*** Inside hashInsert method. Performing hashInsert
```

```
After hashInsert operation ...

HashTable[7]: (dummy,Tania)--->(Tania,NULL)




input is:   %       Aniss

index for data is:  7

linked list before insertion:

HashTable[7]: (dummy,Tania)--->(Tania,NULL)

% is an unrecognizable operation!




input is:   ?       Erik

index for data is:  7

linked list before insertion:

HashTable[7]: (dummy,Tania)--->(Tania,NULL)

** Inside hashRetrieval. Performing hashRetrieval

*** Warning, the record is *not* in the database!




input is:   +       Erik

index for data is:  7

linked list before insertion:

HashTable[7]: (dummy,Tania)--->(Tania,NULL)

*** Inside hashInsert method. Performing hashInsert
```

```
After hashInsert operation ...
HashTable[7]: (dummy,Erik)--->(Erik,Tania)--->(Tania,NULL)



input is:   -       Erik
index for data is:  7
linked list before insertion:
HashTable[7]: (dummy,Erik)--->(Erik,Tania)--->(Tania,NULL)
** Inside hashDelete method. Performing hashDelete
After hashDelete operation ...
HashTable[7]: (dummy,Tania)--->(Tania,NULL)



input is:   ?       Erik
index for data is:  7
linked list before insertion:
HashTable[7]: (dummy,Tania)--->(Tania,NULL)
** Inside hashRetrieval. Performing hashRetrieval
*** Warning, the record is *not* in the database!



input is:   +       Kevin
index for data is:  12
linked list before insertion:
```

```
HashTable[12]: (dummy,Cole)--->(Cole,Daniel)--->(Daniel,NULL)

*** Inside hashInsert method. Performing hashInsert

After hashInsert operation ...

HashTable[12]: (dummy,Cole)--->(Cole,Daniel)--->(Daniel,Kevin)--->(Kevin,NULL)



input is:   +        Arben

index for data is:  10

linked list before insertion:

HashTable[10]: (dummy,NULL)

*** Inside hashInsert method. Performing hashInsert

After hashInsert operation ...

HashTable[10]: (dummy,Arben)--->(Arben,NULL)



input is:   +        David

index for data is:  11

linked list before insertion:

HashTable[11]: (dummy,Jesse)--->(Jesse,NULL)

*** Inside hashInsert method. Performing hashInsert

After hashInsert operation ...

HashTable[11]: (dummy,David)--->(David,Jesse)--->(Jesse,NULL)
```

```
input is:   +       Benjamin

index for data is:  10

linked list before insertion:

HashTable[10]: (dummy,Arben)--->(Arben,NULL)

*** Inside hashInsert method. Performing hashInsert

After hashInsert operation ...

HashTable[10]: (dummy,Arben)--->(Arben,Benjamin)--->(Benjamin,NULL)




input is:   -       Fahim

index for data is:  2

linked list before insertion:

HashTable[2]: (dummy,NULL)

** Inside hashDelete method. Performing hashDelete

*** Warning: data is *not* in the database!




input is:   +       Hengtuo

index for data is:  11

linked list before insertion:

HashTable[11]: (dummy,David)--->(David,Jesse)--->(Jesse,NULL)

*** Inside hashInsert method. Performing hashInsert

After hashInsert operation ...

HashTable[11]: (dummy,David)--->(David,Hengtuo)--->(Hengtuo,Jesse)--->(Jesse,NULL)
```

```
input is:   +        Siliang

index for data is:  0

linked list before insertion:

HashTable[0]: (dummy,Liwen)--->(Liwen,NULL)

*** Inside hashInsert method. Performing hashInsert

After hashInsert operation ...

HashTable[0]: (dummy,Liwen)--->(Liwen,Siliang)--->(Siliang,NULL)




input is:   +        Seungwon

index for data is:  12

linked list before insertion:

HashTable[12]: (dummy,Cole)--->(Cole,Daniel)--->(Daniel,Kevin)--->(Kevin,NULL)

*** Inside hashInsert method. Performing hashInsert

After hashInsert operation ...

HashTable[12]: (dummy,Cole)--->(Cole,Daniel)--->(Daniel,Kevin)--->(Kevin,Seungwon)--->(Seungwon,NULL)




input is:   ?        Yahav

index for data is:  4

linked list before insertion:

HashTable[4]: (dummy,Tsering)--->(Tsering,Yiqing)--->(Yiqing,NULL)
```

```
** Inside hashRetrieval. Performing hashRetrieval

*** Warning, the record is *not* in the database!




input is:   +        Jurgen

index for data is:  10

linked list before insertion:

HashTable[10]: (dummy,Arben)--->(Arben,Benjamin)--->(Benjamin,NULL)

*** Inside hashInsert method. Performing hashInsert

After hashInsert operation ...

HashTable[10]: (dummy,Arben)--->(Arben,Benjamin)--->(Benjamin,Jurgen)--->(Jurgen,NULL)




input is:   -        Gildian

index for data is:  7

linked list before insertion:

HashTable[7]: (dummy,Tania)--->(Tania,NULL)

** Inside hashDelete method. Performing hashDelete

*** Warning: data is *not* in the database!




input is:   ?        Seong

index for data is:  11

linked list before insertion:
```

```
HashTable[11]: (dummy,David)--->(David,Hengtuo)--->(Hengtuo,Jesse)--->(Jesse,NULL)

** Inside hashRetrieval. Performing hashRetrieval

*** Warning, the record is *not* in the database!




input is:   #        Auyon

index for data is:  12

linked list before insertion:

HashTable[12]: (dummy,Cole)--->(Cole,Daniel)--->(Daniel,Kevin)--->(Kevin,Seungwon)--->(Seungwon,NULL)

# is an unrecognizable operation!




input is:   +        Frederick

index for data is:  4

linked list before insertion:

HashTable[4]: (dummy,Tsering)--->(Tsering,Yiqing)--->(Yiqing,NULL)

*** Inside hashInsert method. Performing hashInsert

After hashInsert operation ...

HashTable[4]: (dummy,Frederick)--->(Frederick,Tsering)--->(Tsering,Yiqing)--->(Yiqing,NULL)




input is:   +        Aviraj

index for data is:  11

linked list before insertion:
```

```
HashTable[11]: (dummy,David)--->(David,Hengtuo)--->(Hengtuo,Jesse)--->(Jesse,NULL)

*** Inside hashInsert method. Performing hashInsert

After hashInsert operation ...

HashTable[11]: (dummy,Aviraj)--->(Aviraj,David)--->(David,Hengtuo)--->(Hengtuo,Jesse)---
>(Jesse,NULL)




input is:   +        Shaxzod

index for data is:  12

linked list before insertion:

HashTable[12]: (dummy,Cole)--->(Cole,Daniel)--->(Daniel,Kevin)--->(Kevin,Seungwon)---
>(Seungwon,NULL)

*** Inside hashInsert method. Performing hashInsert

After hashInsert operation ...

HashTable[12]: (dummy,Cole)--->(Cole,Daniel)--->(Daniel,Kevin)--->(Kevin,Seungwon)---
>(Seungwon,Shaxzod)--->(Shaxzod,NULL)




input is:   -        Imran

index for data is:  9

linked list before insertion:

HashTable[9]: (dummy,Mohammad)--->(Mohammad,NULL)

** Inside hashDelete method. Performing hashDelete

*** Warning: data is *not* in the database!
```

```
input is:   +       Zai
index for data is:  9
linked list before insertion:
HashTable[9]: (dummy,Mohammad)--->(Mohammad,NULL)
*** Inside hashInsert method. Performing hashInsert
After hashInsert operation ...
HashTable[9]: (dummy,Mohammad)--->(Mohammad,Zai)--->(Zai,NULL)



input is:   ?       Jonathan
index for data is:  3
linked list before insertion:
HashTable[3]: (dummy,NULL)
** Inside hashRetrieval. Performing hashRetrieval
*** Warning, the record is *not* in the database!



input is:   +       Sean
index for data is:  9
linked list before insertion:
HashTable[9]: (dummy,Mohammad)--->(Mohammad,Zai)--->(Zai,NULL)
*** Inside hashInsert method. Performing hashInsert
After hashInsert operation ...
HashTable[9]: (dummy,Mohammad)--->(Mohammad,Sean)--->(Sean,Zai)--->(Zai,NULL)
```

```
input is:   -        Brandon

index for data is:  6

linked list before insertion:

HashTable[6]: (dummy,Jia)--->(Jia,Manana)--->(Manana,Ryan)--->(Ryan,Tshetrim)---
>(Tshetrim,NULL)

** Inside hashDelete method. Performing hashDelete

*** Warning: data is *not* in the database!




input is:   ?        Juan

index for data is:  6

linked list before insertion:

HashTable[6]: (dummy,Jia)--->(Jia,Manana)--->(Manana,Ryan)--->(Ryan,Tshetrim)---
>(Tshetrim,NULL)

** Inside hashRetrieval. Performing hashRetrieval

*** Warning, the record is *not* in the database!




input is:   +        Steven

index for data is:  0

linked list before insertion:

HashTable[0]: (dummy,Liwen)--->(Liwen,Siliang)--->(Siliang,NULL)

*** Inside hashInsert method. Performing hashInsert
```

```
After hashInsert operation ...

HashTable[0]: (dummy,Liwen)--->(Liwen,Siliang)--->(Siliang,Steven)--->(Steven,NULL)




input is:   +        Tim

index for data is:  1

linked list before insertion:

HashTable[1]: (dummy,NULL)

*** Inside hashInsert method. Performing hashInsert

After hashInsert operation ...

HashTable[1]: (dummy,Tim)--->(Tim,NULL)




input is:   +        Matthew

index for data is:  0

linked list before insertion:

HashTable[0]: (dummy,Liwen)--->(Liwen,Siliang)--->(Siliang,Steven)--->(Steven,NULL)

*** Inside hashInsert method. Performing hashInsert

After hashInsert operation ...

HashTable[0]: (dummy,Liwen)--->(Liwen,Matthew)--->(Matthew,Siliang)--->(Siliang,Steven)---
>(Steven,NULL)




input is:   +        Natasha

index for data is:  5
```

```
linked list before insertion:

HashTable[5]: (dummy,NULL)

*** Inside hashInsert method. Performing hashInsert

After hashInsert operation ...

HashTable[5]: (dummy,Natasha)--->(Natasha,NULL)




input is:   -         Natasha

index for data is:  5

linked list before insertion:

HashTable[5]: (dummy,Natasha)--->(Natasha,NULL)

** Inside hashDelete method. Performing hashDelete

After hashDelete operation ...

HashTable[5]: (dummy,NULL)




input is:   ?         Natasha

index for data is:  5

linked list before insertion:

HashTable[5]: (dummy,NULL)

** Inside hashRetrieval. Performing hashRetrieval

*** Warning, the record is *not* in the database!
```