**IMPLEMENTATION *OF ARRAYLIST AND HASHMAP***

**Name: Aditya Aggarwal Roll Number: IIT2019210**

**1) ArrayList Implementation**

For this I have a created a separate class named MyArrayList and implemented some methods of original ArrayList and there is also some variation in some of the methods, example in method set: an index and a value is passed but ArrayList will show error if index is out of range and MyArrayList will round the index to index by index=index% size and will not show error and proceed and I have used Array to implement MyArrayList.

Code of class MyArrayList:

package mypac;  
  
public class MyArrayList<V> {  
  
 //Since we are implementing arraylist using array so its fix\_size has to be defined  
 private static final int *fix\_size*=10;  
  
 //recordArray is a array of Object type  
 private Object recordArray[];  
  
 //index is the original size of arraylist and size is the max size of recordArray  
 private int index,size;  
  
 //Default Constructor  
 public MyArrayList(){  
 recordArray=new Object[*fix\_size*];  
 this.size=*fix\_size*;  
 }  
  
 //Method to add an element in the recordArray which will also check size bound of array  
 public void add(V v){  
 if(this.index==this.size-1){  
 //we need to increase the size of recordArray[]  
 increaseSize();  
 }  
 recordArray[this.index]=v;  
 this.index++;  
  
 }  
  
 //Private method to handle size bound of the array  
 private void increaseSize() {  
 this.size=this.size+*fix\_size*;  
 Object newrecordArray[]=new Object[this.size];  
 for(int i=0; i<recordArray.length;i++){  
 newrecordArray[i]=recordArray[i];  
 }  
 this.recordArray=newrecordArray;  
 }  
  
 //Method to get the element of recordArray at ith index  
 //Note: Different from original ArrayList since it will also work for indexes out of range  
 public V get(int i){  
 i=i%index;  
 final V v = (V)recordArray[i];  
 return v;  
  
 }  
  
 //Method to remove an element of recordArray from ith index\  
 //Note: Different from original ArrayList since it will also work for indexes out of range  
 public void remove(int i){  
 i=i%index;  
 for(int x=i; x<this.recordArray.length-1;x++){  
 recordArray[x]=recordArray[x+1];  
 }  
 this.index--;  
 }  
  
 //Method to set the value of element at given index  
 //Note: Different from original ArrayList since it will also work for indexes out of range  
 public void set(Integer i,V v){  
 i=i%index;  
 recordArray[i]=v;  
 }  
  
 //Method to clear all the elements in the array  
 public void clear(){  
 index=0;  
 size=*fix\_size*;  
 }  
  
 //Method to return the size of ArrayList  
 public int size(){  
 return index;  
 }  
  
 //Method to display all the elements in the ArrayList  
 public void display(){  
 for(int i=0;i<this.index;i++){  
 System.*out*.print(recordArray[i]);  
 if(i!=this.index-1)  
 System.*out*.print(", ");  
 }  
 System.*out*.println();  
 }  
}

After this I have made a test class in the same package to test the working of all the methods, so I have used a switch to switch between the methods and through it we can test every method as many numbers of times.

Object is people and we are implementing MyArrayList on their name.

Code of class TestClass:

package mypac;  
import mypac.MyArrayList;  
  
import java.util.Scanner;  
  
public class TestClass{  
 public static void main(String[] args) {  
 System.*out*.println("Implementation of ArrayList:");  
 //Let us take an ArrayList of people with their name as the value.  
 MyArrayList<String> people = new MyArrayList<String>();  
 boolean flag=true;  
 while(flag){  
 System.*out*.println();  
 System.*out*.println("Enter 1 to add a person at the end");  
 System.*out*.println("Enter 2 to set name of a person at the given index");  
 System.*out*.println("Enter 3 to print name of a person at given index");  
 System.*out*.println("Enter 4 to remove a person from the ArrayList");  
 System.*out*.println("Enter 5 to clear the whole ArrayList");  
 System.*out*.println("Enter 6 to print the size of ArrayList");  
 System.*out*.println("Enter 7 to print the whole ArrayList");  
 System.*out*.println("Enter 8 to exit");  
 Scanner sc=new Scanner(System.*in*);  
 int x=sc.nextInt();  
 String name;  
 Integer index;  
 switch(x){  
 case 1: System.*out*.printf("Enter name of person: ");  
 name=sc.next();  
 people.add(name);  
 System.*out*.println("Person is added with name: " + name);  
 break;  
 case 2: System.*out*.printf("Enter name of person: ");  
 name=sc.next();  
 System.*out*.printf("Enter index where this person is to be added: ");  
 index=sc.nextInt();  
 people.set(index, name);  
 System.*out*.printf("Person at index " + index + " is: " + people.get(index));  
 break;  
 case 3: System.*out*.printf("Enter index from where person name is to be displayed: ");  
 index=sc.nextInt();  
 System.*out*.println("Person at index " + index + " is: " + people.get(index));  
 break;  
 case 4: System.*out*.printf("Enter index from where person is to be removed: ");  
 index=sc.nextInt();  
 people.remove(index);  
 System.*out*.println("Person at index " + index + " is successfully removed from the arraylist");  
 break;  
 case 5: people.clear();  
 System.*out*.println("ArrayList is cleared");  
 break;  
 case 6: if(people.size()==0) {  
 System.*out*.println("No person in the arraylist");  
 break;  
 }  
 System.*out*.println("Number of persons in arraylist are: " + people.size());  
 break;  
 case 7: if(people.size()==0) {  
 System.*out*.println("No person in the ArrayList");  
 break;  
 }  
 System.*out*.printf("The ArrayList is: ");  
 people.display();  
 break;  
 case 8: flag=false;  
 break;  
 default: System.*out*.println("Not a valid option, try another option!!!");  
 }  
 }  
 }  
}

Sample Output:

Implementation of ArrayList:  
  
 Enter 1 to add a person at the end  
 Enter 2 to set name of a person at the given index  
 Enter 3 to print name of a person at given index  
 Enter 4 to remove a person from the ArrayList  
 Enter 5 to clear the whole ArrayList  
 Enter 6 to print the size of ArrayList  
 Enter 7 to print the whole ArrayList  
 Enter 8 to exit  
 1  
 Enter name of person: aditya  
 Person is added with name: aditya  
  
 Enter 1 to add a person at the end  
 Enter 2 to set name of a person at the given index  
 Enter 3 to print name of a person at given index  
 Enter 4 to remove a person from the ArrayList  
 Enter 5 to clear the whole ArrayList  
 Enter 6 to print the size of ArrayList  
 Enter 7 to print the whole ArrayList  
 Enter 8 to exit  
 1  
 Enter name of person: mayank  
 Person is added with name: mayank  
  
 Enter 1 to add a person at the end  
 Enter 2 to set name of a person at the given index  
 Enter 3 to print name of a person at given index  
 Enter 4 to remove a person from the ArrayList  
 Enter 5 to clear the whole ArrayList  
 Enter 6 to print the size of ArrayList  
 Enter 7 to print the whole ArrayList  
 Enter 8 to exit  
 6  
 Number of persons in arraylist are: 2  
  
 Enter 1 to add a person at the end  
 Enter 2 to set name of a person at the given index  
 Enter 3 to print name of a person at given index  
 Enter 4 to remove a person from the ArrayList  
 Enter 5 to clear the whole ArrayList  
 Enter 6 to print the size of ArrayList  
 Enter 7 to print the whole ArrayList  
 Enter 8 to exit  
 7  
 The ArrayList is: aditya, mayank  
  
 Enter 1 to add a person at the end  
 Enter 2 to set name of a person at the given index  
 Enter 3 to print name of a person at given index  
 Enter 4 to remove a person from the ArrayList  
 Enter 5 to clear the whole ArrayList  
 Enter 6 to print the size of ArrayList  
 Enter 7 to print the whole ArrayList  
 Enter 8 to exit  
 2  
 Enter name of person: mohan  
 Enter index where this person is to be added: 1  
 Person at index 1 is: mohan  
 Enter 1 to add a person at the end  
 Enter 2 to set name of a person at the given index  
 Enter 3 to print name of a person at given index  
 Enter 4 to remove a person from the ArrayList  
 Enter 5 to clear the whole ArrayList  
 Enter 6 to print the size of ArrayList  
 Enter 7 to print the whole ArrayList  
 Enter 8 to exit  
 3  
 Enter index from where person name is to be displayed: 1  
 Person at index 1 is: mohan  
  
 Enter 1 to add a person at the end  
 Enter 2 to set name of a person at the given index  
 Enter 3 to print name of a person at given index  
 Enter 4 to remove a person from the ArrayList  
 Enter 5 to clear the whole ArrayList  
 Enter 6 to print the size of ArrayList  
 Enter 7 to print the whole ArrayList  
 Enter 8 to exit  
 7  
 The ArrayList is: aditya, mohan  
  
 Enter 1 to add a person at the end  
 Enter 2 to set name of a person at the given index  
 Enter 3 to print name of a person at given index  
 Enter 4 to remove a person from the ArrayList  
 Enter 5 to clear the whole ArrayList  
 Enter 6 to print the size of ArrayList  
 Enter 7 to print the whole ArrayList  
 Enter 8 to exit  
 4  
 Enter index from where person is to be removed: 1  
 Person at index 1 is successfully removed from the arraylist  
  
 Enter 1 to add a person at the end  
 Enter 2 to set name of a person at the given index  
 Enter 3 to print name of a person at given index  
 Enter 4 to remove a person from the ArrayList  
 Enter 5 to clear the whole ArrayList  
 Enter 6 to print the size of ArrayList  
 Enter 7 to print the whole ArrayList  
 Enter 8 to exit  
 7  
 The ArrayList is: aditya  
  
 Enter 1 to add a person at the end  
 Enter 2 to set name of a person at the given index  
 Enter 3 to print name of a person at given index  
 Enter 4 to remove a person from the ArrayList  
 Enter 5 to clear the whole ArrayList  
 Enter 6 to print the size of ArrayList  
 Enter 7 to print the whole ArrayList  
 Enter 8 to exit  
 9  
 Not a valid option, try another option!!!  
  
 Enter 1 to add a person at the end  
 Enter 2 to set name of a person at the given index  
 Enter 3 to print name of a person at given index  
 Enter 4 to remove a person from the ArrayList  
 Enter 5 to clear the whole ArrayList  
 Enter 6 to print the size of ArrayList  
 Enter 7 to print the whole ArrayList  
 Enter 8 to exit  
 5  
 ArrayList is cleared  
  
 Enter 1 to add a person at the end  
 Enter 2 to set name of a person at the given index  
 Enter 3 to print name of a person at given index  
 Enter 4 to remove a person from the ArrayList  
 Enter 5 to clear the whole ArrayList  
 Enter 6 to print the size of ArrayList  
 Enter 7 to print the whole ArrayList  
 Enter 8 to exit  
 7  
 No person in the ArrayList  
  
 Enter 1 to add a person at the end  
 Enter 2 to set name of a person at the given index  
 Enter 3 to print name of a person at given index  
 Enter 4 to remove a person from the ArrayList  
 Enter 5 to clear the whole ArrayList  
 Enter 6 to print the size of ArrayList  
 Enter 7 to print the whole ArrayList  
 Enter 8 to exit  
 6  
 No person in the arraylist  
  
 Enter 1 to add a person at the end  
 Enter 2 to set name of a person at the given index  
 Enter 3 to print name of a person at given index  
 Enter 4 to remove a person from the ArrayList  
 Enter 5 to clear the whole ArrayList  
 Enter 6 to print the size of ArrayList  
 Enter 7 to print the whole ArrayList  
 Enter 8 to exit  
 8  
  
 Process finished with exit code 0

**2) HashMap Implementation**

For this I have a created a separate class named MyHashMap and implemented some methods of original HashMap and there is also some variation in some of the methods like in MyArrayList, I have used ArrayList to implement MyHashMap.

<K,V> is used so that user can take any data type of his/her choice.

Code of class MyHashMap:

package mypack;  
  
import java.util.ArrayList;  
import java.util.List;  
public class MyHashMap<K,V>{  
  
 //This class will contain key and value pair for every element.  
 class Container{  
 K key;  
 V value;  
 public void insert(K k, V v){  
 this.key=k;  
 this.value=v;  
 }  
 }  
   
 private Container c;  
 //recordList is a list of type Container, for ith index it will give value of key and pair.  
 private List<Container> recordList;  
  
 //Default Constructor to define the recordList.  
 public MyHashMap(){  
   
 this.recordList=new ArrayList<Container>();  
 }  
  
 //put method will add a key and its value in recordList and will also check for duplicate keys.  
 public void put(K k, V v){  
 this.c=new Container();  
 c.insert(k, v);  
 int i;  
 //It will check for the same key before adding  
 for(i=0; i<recordList.size(); i++){  
 Container c1=recordList.get(i);  
 if(c1.key.equals(k)){  
 //This will remove the existing key, value pair  
 recordList.remove(i);  
 break;  
 }  
 }  
 //To add the new key, value at index i  
 recordList.add(i,c);  
 }  
  
 //This method will return the value for the given key.  
 public V get(K k) {  
 for (int i = 0; i < this.recordList.size(); i++) {  
 Container con = recordList.get(i);  
 if(con.key.equals(k)){  
 return con.value;  
 }  
  
 }  
 return null;  
 }  
  
 //This method will delete key, value pair if given key exists and returns true else it will return false  
 public boolean remove(K k){  
 for (int i = 0; i < this.recordList.size(); i++) {  
 Container con = recordList.get(i);  
 if(con.key.equals(k)){  
 //remove the existing String  
 recordList.remove(i);  
 return true;  
 }  
  
 }  
 return false;  
 }  
  
 //This method will empty the hashMap array  
 public void clear(){  
 for (int i = this.recordList.size() - 1; i >= 0; i--) {  
 Container con = recordList.get(i);  
 recordList.remove(i);  
 }  
 }  
  
 //This method will return the number of different keys in the recordList  
 public int size(){  
 return recordList.size();  
 }  
  
 //This method displays all the keys and their respective values.  
 public void display(){  
 for(int i=0;i<this.recordList.size();i++){  
 Container con = recordList.get(i);  
 System.*out*.print(con.key + " -> " + con.value);  
 if(i!=this.recordList.size()-1)  
 System.*out*.print(", ");  
 }  
 System.*out*.println();  
 }  
  
 //This method return Key at the given index.  
 //If index is out of range then it will also handle that case.  
 public K getKey(int index){  
 if(recordList.size()==0)  
 return null;  
 index=index%(this.recordList.size());  
 Container con = recordList.get(index);  
 return con.key;  
 }  
  
 //This method return Value at the given index.  
 //If index is out of range then it will also handle that case.  
 public V getValue(int index){  
 if(recordList.size()==0)  
 return null;  
 index=index%(this.recordList.size());  
 Container con = recordList.get(index);  
 return con.value;  
 }  
}

After this I have made a test class in the same package to test the working of all the methods, so I have used a switch to switch between the methods and through it we can test every method as many numbers of times.

Object is people and we are implementing MyHashMap on their name as key and age as value.

Code of class TestClass:

package mypack;  
import mypack.MyHashMap;  
  
import java.util.Scanner;  
  
public class TestClass{  
 public static void main(String[] args) {  
 System.*out*.println("Implementation of HashMap:");  
 //Let us take an HashMap of people with their name as Key and age as Value.  
 MyHashMap<String, Integer> people = new MyHashMap<String, Integer>();  
 boolean flag=true;  
 while(flag){  
 System.*out*.println();  
 System.*out*.println("Enter 1 to add a person");  
 System.*out*.println("Enter 2 to print age of a person");  
 System.*out*.println("Enter 3 to remove a person from the hashmap");  
 System.*out*.println("Enter 4 to clear the whole hashmap");  
 System.*out*.println("Enter 5 to print the size of hashmap");  
 System.*out*.println("Enter 6 to print the whole hashmap");  
 System.*out*.println("Enter 7 to print key at a specific index");  
 System.*out*.println("Enter 8 to print value at a specific index");  
 System.*out*.println("Enter 9 to print all keys using a loop");  
 System.*out*.println("Enter 10 to print all values using a loop");  
 System.*out*.println("Enter 11 to exit");  
 Scanner sc=new Scanner(System.*in*);  
 int x=sc.nextInt();  
 String name;  
 Integer age,index;  
 switch(x){  
 case 1: System.*out*.printf("Enter name of person: ");  
 name=sc.next();  
 System.*out*.printf("Enter age of person: ");  
 age=sc.nextInt();  
 people.put(name,age);  
 System.*out*.println("Person is added with name: " + name + " and age: " + age);  
 break;  
 case 2: System.*out*.printf("Enter name of person whose age is to be displayed: ");  
 name=sc.next();  
 if(people.get(name)==null)  
 System.*out*.println("No such person in hashmap");  
 else  
 System.*out*.println("Age of " + name + " is " + people.get(name));  
 break;  
 case 3: System.*out*.printf("Enter name of person to be removed: ");  
 name=sc.next();  
 boolean var=people.remove(name);  
 if(!var)  
 System.*out*.println("No such person in hashmap");  
 else  
 System.*out*.println(name + " is successfully removed from the hashmap");  
 break;  
 case 4: people.clear();  
 System.*out*.println("HashMap is cleared");  
 break;  
 case 5: if(people.size()==0) {  
 System.*out*.println("No person in the HashMap");  
 break;  
 }  
 System.*out*.println("Number of (key, value) pairs in hashmap are: " + people.size());  
 break;  
 case 6: if(people.size()==0) {  
 System.*out*.println("No person in the HashMap");  
 break;  
 }  
 System.*out*.printf("The HashMap is: ");  
 people.display();  
 break;  
 case 7: if(people.size()==0) {  
 System.*out*.println("No person in the HashMap");  
 break;  
 }  
 System.*out*.printf("Enter specific index at which name would be displayed: ");  
 index=sc.nextInt();  
 System.*out*.println("Name at index " + index + " is: " + people.getKey(index));  
 break;  
 case 8: if(people.size()==0) {  
 System.*out*.println("No person in the HashMap");  
 break;  
 }  
 System.*out*.printf("Enter specific index at which age would be displayed: ");  
 index=sc.nextInt();  
 System.*out*.println("Age at index " + index + " is: " + people.getValue(index));  
 break;  
 case 9: if(people.size()==0) {  
 System.*out*.println("No person in the HashMap");  
 break;  
 }  
 System.*out*.printf("Names are: ");  
 for(int i=0;i<people.size();i++) {  
 System.*out*.printf(people.getKey(i));  
 if(i!=people.size()-1)  
 System.*out*.printf(", ");  
 }  
 break;  
 case 10: if(people.size()==0) {  
 System.*out*.println("No person in the HashMap");  
 break;  
 }  
 System.*out*.printf("Ages are: ");  
 for(int i=0;i<people.size();i++) {  
 System.*out*.print(people.getValue(i));  
 if(i!=people.size()-1)  
 System.*out*.printf(", ");  
 }  
 break;  
 case 11: flag=false;  
 break;  
 default: System.*out*.println("Not a valid option, try another option!!!");  
 }  
 }  
 }  
}

Sample Output:

Implementation of HashMap:  
  
 Enter 1 to add a person  
 Enter 2 to print age of a person  
 Enter 3 to remove a person from the hashmap  
 Enter 4 to clear the whole hashmap  
 Enter 5 to print the size of hashmap  
 Enter 6 to print the whole hashmap  
 Enter 7 to print key at a specific index  
 Enter 8 to print value at a specific index  
 Enter 9 to print all keys using a loop  
 Enter 10 to print all values using a loop  
 Enter 11 to exit  
 1  
 Enter name of person: aditya  
 Enter age of person: 19  
 Person is added with name: aditya and age: 19  
  
 Enter 1 to add a person  
 Enter 2 to print age of a person  
 Enter 3 to remove a person from the hashmap  
 Enter 4 to clear the whole hashmap  
 Enter 5 to print the size of hashmap  
 Enter 6 to print the whole hashmap  
 Enter 7 to print key at a specific index  
 Enter 8 to print value at a specific index  
 Enter 9 to print all keys using a loop  
 Enter 10 to print all values using a loop  
 Enter 11 to exit  
 1  
 Enter name of person: mayank  
 Enter age of person: 17  
 Person is added with name: mayank and age: 17  
  
 Enter 1 to add a person  
 Enter 2 to print age of a person  
 Enter 3 to remove a person from the hashmap  
 Enter 4 to clear the whole hashmap  
 Enter 5 to print the size of hashmap  
 Enter 6 to print the whole hashmap  
 Enter 7 to print key at a specific index  
 Enter 8 to print value at a specific index  
 Enter 9 to print all keys using a loop  
 Enter 10 to print all values using a loop  
 Enter 11 to exit  
 2  
 Enter name of person whose age is to be displayed: aditya  
 Age of aditya is 19  
  
 Enter 1 to add a person  
 Enter 2 to print age of a person  
 Enter 3 to remove a person from the hashmap  
 Enter 4 to clear the whole hashmap  
 Enter 5 to print the size of hashmap  
 Enter 6 to print the whole hashmap  
 Enter 7 to print key at a specific index  
 Enter 8 to print value at a specific index  
 Enter 9 to print all keys using a loop  
 Enter 10 to print all values using a loop  
 Enter 11 to exit  
 2  
 Enter name of person whose age is to be displayed: adityaa  
 No such person in hashmap  
  
 Enter 1 to add a person  
 Enter 2 to print age of a person  
 Enter 3 to remove a person from the hashmap  
 Enter 4 to clear the whole hashmap  
 Enter 5 to print the size of hashmap  
 Enter 6 to print the whole hashmap  
 Enter 7 to print key at a specific index  
 Enter 8 to print value at a specific index  
 Enter 9 to print all keys using a loop  
 Enter 10 to print all values using a loop  
 Enter 11 to exit  
 5  
 Number of (key, value) pairs in hashmap are: 2  
  
 Enter 1 to add a person  
 Enter 2 to print age of a person  
 Enter 3 to remove a person from the hashmap  
 Enter 4 to clear the whole hashmap  
 Enter 5 to print the size of hashmap  
 Enter 6 to print the whole hashmap  
 Enter 7 to print key at a specific index  
 Enter 8 to print value at a specific index  
 Enter 9 to print all keys using a loop  
 Enter 10 to print all values using a loop  
 Enter 11 to exit  
 6  
 The HashMap is: aditya -> 19, mayank -> 17  
  
 Enter 1 to add a person  
 Enter 2 to print age of a person  
 Enter 3 to remove a person from the hashmap  
 Enter 4 to clear the whole hashmap  
 Enter 5 to print the size of hashmap  
 Enter 6 to print the whole hashmap  
 Enter 7 to print key at a specific index  
 Enter 8 to print value at a specific index  
 Enter 9 to print all keys using a loop  
 Enter 10 to print all values using a loop  
 Enter 11 to exit  
 7  
 Enter specific index at which name would be displayed: 0  
 Name at index 0 is: aditya  
  
 Enter 1 to add a person  
 Enter 2 to print age of a person  
 Enter 3 to remove a person from the hashmap  
 Enter 4 to clear the whole hashmap  
 Enter 5 to print the size of hashmap  
 Enter 6 to print the whole hashmap  
 Enter 7 to print key at a specific index  
 Enter 8 to print value at a specific index  
 Enter 9 to print all keys using a loop  
 Enter 10 to print all values using a loop  
 Enter 11 to exit  
 8  
 Enter specific index at which age would be displayed: 1  
 Age at index 1 is: 17  
  
 Enter 1 to add a person  
 Enter 2 to print age of a person  
 Enter 3 to remove a person from the hashmap  
 Enter 4 to clear the whole hashmap  
 Enter 5 to print the size of hashmap  
 Enter 6 to print the whole hashmap  
 Enter 7 to print key at a specific index  
 Enter 8 to print value at a specific index  
 Enter 9 to print all keys using a loop  
 Enter 10 to print all values using a loop  
 Enter 11 to exit  
 9  
 Names are: aditya, mayank  
 Enter 1 to add a person  
 Enter 2 to print age of a person  
 Enter 3 to remove a person from the hashmap  
 Enter 4 to clear the whole hashmap  
 Enter 5 to print the size of hashmap  
 Enter 6 to print the whole hashmap  
 Enter 7 to print key at a specific index  
 Enter 8 to print value at a specific index  
 Enter 9 to print all keys using a loop  
 Enter 10 to print all values using a loop  
 Enter 11 to exit  
 10  
 Ages are: 19, 17  
 Enter 1 to add a person  
 Enter 2 to print age of a person  
 Enter 3 to remove a person from the hashmap  
 Enter 4 to clear the whole hashmap  
 Enter 5 to print the size of hashmap  
 Enter 6 to print the whole hashmap  
 Enter 7 to print key at a specific index  
 Enter 8 to print value at a specific index  
 Enter 9 to print all keys using a loop  
 Enter 10 to print all values using a loop  
 Enter 11 to exit  
 12  
 Not a valid option, try another option!!!  
  
 Enter 1 to add a person  
 Enter 2 to print age of a person  
 Enter 3 to remove a person from the hashmap  
 Enter 4 to clear the whole hashmap  
 Enter 5 to print the size of hashmap  
 Enter 6 to print the whole hashmap  
 Enter 7 to print key at a specific index  
 Enter 8 to print value at a specific index  
 Enter 9 to print all keys using a loop  
 Enter 10 to print all values using a loop  
 Enter 11 to exit  
 3  
 Enter name of person to be removed: may  
 No such person in hashmap  
  
 Enter 1 to add a person  
 Enter 2 to print age of a person  
 Enter 3 to remove a person from the hashmap  
 Enter 4 to clear the whole hashmap  
 Enter 5 to print the size of hashmap  
 Enter 6 to print the whole hashmap  
 Enter 7 to print key at a specific index  
 Enter 8 to print value at a specific index  
 Enter 9 to print all keys using a loop  
 Enter 10 to print all values using a loop  
 Enter 11 to exit  
 3  
 Enter name of person to be removed: mayank  
 mayank is successfully removed from the hashmap  
  
 Enter 1 to add a person  
 Enter 2 to print age of a person  
 Enter 3 to remove a person from the hashmap  
 Enter 4 to clear the whole hashmap  
 Enter 5 to print the size of hashmap  
 Enter 6 to print the whole hashmap  
 Enter 7 to print key at a specific index  
 Enter 8 to print value at a specific index  
 Enter 9 to print all keys using a loop  
 Enter 10 to print all values using a loop  
 Enter 11 to exit  
 5  
 Number of (key, value) pairs in hashmap are: 1  
  
 Enter 1 to add a person  
 Enter 2 to print age of a person  
 Enter 3 to remove a person from the hashmap  
 Enter 4 to clear the whole hashmap  
 Enter 5 to print the size of hashmap  
 Enter 6 to print the whole hashmap  
 Enter 7 to print key at a specific index  
 Enter 8 to print value at a specific index  
 Enter 9 to print all keys using a loop  
 Enter 10 to print all values using a loop  
 Enter 11 to exit  
 6  
 The HashMap is: aditya -> 19  
  
 Enter 1 to add a person  
 Enter 2 to print age of a person  
 Enter 3 to remove a person from the hashmap  
 Enter 4 to clear the whole hashmap  
 Enter 5 to print the size of hashmap  
 Enter 6 to print the whole hashmap  
 Enter 7 to print key at a specific index  
 Enter 8 to print value at a specific index  
 Enter 9 to print all keys using a loop  
 Enter 10 to print all values using a loop  
 Enter 11 to exit  
 4  
 HashMap is cleared  
  
 Enter 1 to add a person  
 Enter 2 to print age of a person  
 Enter 3 to remove a person from the hashmap  
 Enter 4 to clear the whole hashmap  
 Enter 5 to print the size of hashmap  
 Enter 6 to print the whole hashmap  
 Enter 7 to print key at a specific index  
 Enter 8 to print value at a specific index  
 Enter 9 to print all keys using a loop  
 Enter 10 to print all values using a loop  
 Enter 11 to exit  
 6  
 No person in the HashMap  
  
 Enter 1 to add a person  
 Enter 2 to print age of a person  
 Enter 3 to remove a person from the hashmap  
 Enter 4 to clear the whole hashmap  
 Enter 5 to print the size of hashmap  
 Enter 6 to print the whole hashmap  
 Enter 7 to print key at a specific index  
 Enter 8 to print value at a specific index  
 Enter 9 to print all keys using a loop  
 Enter 10 to print all values using a loop  
 Enter 11 to exit  
 11  
  
 Process finished with exit code 0