**Lab 12**

**Question 01:**

import java.util.\*;

class Question\_number1{

int[] arr;

int mod;

Question\_number1(int data){

arr = new int[data];

mod = data;

}

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

System.out.println("Enter limit of hash table: ");

int n = sc.nextInt();

Question\_number1 q = new Question\_number1(n);

System.out.println("Enter values: ");

for (int i = 0; i < n; i++) {

q.division\_method(sc.nextInt());

}

q.print();

System.out.println("\n----------------------------------");

System.out.println("For searching: ");

System.out.println("Enter number that you wanna search: ");

q.searching(sc.nextInt());

System.out.println("\n----------------------------------");

System.out.println("For deletion: ");

System.out.println("Enter value that you wanna delete: ");

q.delete(sc.nextInt());

q.print();

}

// ------------------------Deletion---------------------------

public void delete(int data){

if (arr[data%mod]==data){

arr[data%mod]=-1;

}

}

// -------------------------Searching-------------------

public void searching(int find){

if (arr[find%mod]==find) {

System.out.println("Number Found in " + find % mod + " position");

}else

System.out.println("Number is not present: ");

}

public void print(){

for (int i = 0; i < arr.length; i++) {

System.out.print(arr[i]+"->");

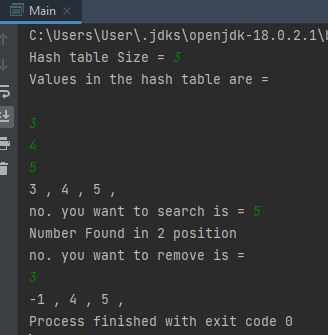
} }

public void division\_method(int data){

int num = data % mod ;

arr[num] = data;

} }



**Question 02:**

public class Qno2\_Lab12{

public static void main(String[] args){

Hashing hg = new Hashing();

hg.Initial();

System.out.println("\*\* Linear\_Probing \*\*");

hg.Linear\_Probing(3);

hg.Linear\_Probing(2);

hg.Linear\_Probing(9);

hg.Linear\_Probing(6);

hg.Linear\_Probing(11);

hg.Linear\_Probing(13);

hg.Linear\_Probing(7);

hg.Linear\_Probing(12);

hg.Print();

hg.Initial();

System.out.println("\*\* Quadratic\_Probing \*\*");

hg.Quadratic\_Probing(3);

hg.Quadratic\_Probing(2);

hg.Quadratic\_Probing(9);

hg.Quadratic\_Probing(6);

hg.Quadratic\_Probing(11);

hg.Quadratic\_Probing(13);

hg.Quadratic\_Probing(7);

hg.Quadratic\_Probing(12);

hg.Print();

}

}

class Hashing{

int arr[] = new int[10];

int size=arr.length;

public void Initial(){

for(int i=0;i<size;i++){

arr[i]=-1;

}

}

public void Linear\_Probing(int data){

int key=data%size;

if(arr[key]==-1){

arr[key]=data;

}

else{

for(int i=1;i<size;i++){

int num=(key+i)%size;

if(arr[num]==-1) {

arr[num]=data;

break;

}

}

}

}

public void Quadratic\_Probing(int data){

int key=data%size;

if(arr[key]==-1){

arr[key]=data;

}

else{

for(int i=1;i<size;i++){

int num=(key+(i\*i))%size;

if(arr[num]==-1){

arr[num]=data;

break;

}

}

}

}

public void Print(){

for(int i=0;i<size;i++){

if(arr[i]!=-1){

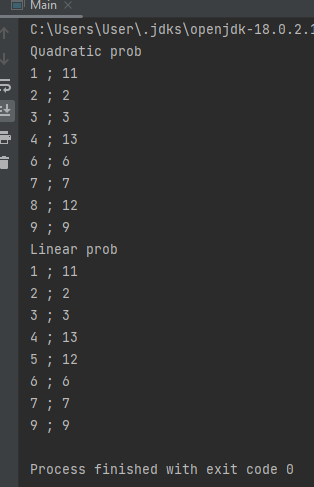
System.out.println("Index: "+i+" | Value: "+arr[i]);

}

}

}

}



**Question 03:**

import java.util.\*;

public class Lab12Task3 {

class ContactBook{

String studentId;

int contactNumber;

ContactBook next;

ContactBook(String studentId,int contactNumber){

this.studentId=studentId;

this.contactNumber=contactNumber;

this.next=null;

}

}

public int tableSize;

public ContactBook[] book;

public int size=0;

Lab12Task3(int tableSize){

size=0;

this.tableSize=tableSize;

book=new ContactBook[tableSize];

for (int i=0;i<tableSize;i++){

book[i]=null;

}

}

public int getSize(){

return size;

}

public boolean isEmpty(){

if (size==0){

return true;

}

return false;

}

public int getPair(String studentId){

int value=studentId.hashCode()%10;

if (book[value]==null){

return -1;

}

else {

ContactBook book1=book[value];

while (book1!=null&&!book1.studentId.equals(studentId)){

book1=book1.next;

}

if (book1==null){

return -1;

}

else {

return book1.contactNumber;

}

}

}

public void insert(String studentId,int contactNumber){

int value=studentId.hashCode()%10;

if (book[value]==null){

book[value]=new ContactBook(studentId,contactNumber);

}

else {

ContactBook book1=book[value];

while (book1.next != null && !book1.studentId.equals(studentId)) {

book1 = book1.next;

}

if (book1.studentId.equals(studentId)) {

book1.contactNumber = contactNumber;

}

else {

book1.next = new ContactBook(studentId, contactNumber);

}

}

size++;

}

public void delete(String studentId){

int value=studentId.hashCode()%10;

ContactBook prev=null;

ContactBook book1=book[value];

if (book[value]!=null){

while (book1.next!=null&&!book1.studentId.equals(studentId)){

prev=book1;

book1=book1.next;

}

if (book1.studentId.equals(studentId)){

if (prev==null){

book[value]=book1.next;

}

else {

prev.next=book1.next;

}

size--;

}

}

}

public void print() {

for (int i = 0; i < tableSize; i++) {

ContactBook book1 = book[i];

while (book1 != null) {

System.out.println("Student ID = " + book1.studentId + " " + "Contact Number = " + book1.contactNumber+" at index "+i);

book1 = book1.next;

}

}

System.out.println();

}

public static void main(String[] args) {

Lab12Task3 hash=new Lab12Task3(10);

System.out.println(hash.isEmpty());

System.out.println("Contact Book");

hash.insert("3803",3024564);

hash.insert("3813",3057356);

hash.insert("3913",3029246);

hash.insert("3807",3022854);

System.out.println("Contact book size is: "+hash.getSize());

hash.print();

System.out.println("The contact number of 3803 is "+hash.getPair("3803"));

System.out.println();

System.out.println("Deleting a contact number");

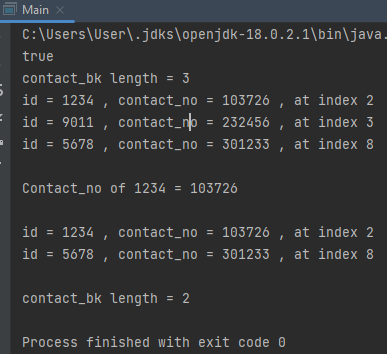
hash.delete("3807");

hash.print();

System.out.println();

System.out.println("Contact book size is: "+hash.getSize());

} }



**Question 05:**

import java.util.\*;

class node{

String password;

String name;

node next;

node(String password,String name){

this.password=password;

this.name=name;

this.next=null;

}

}

public class Lab12Task5 {

node[] hashing;

int size;

Lab12Task5(int size){

this.size=size;

hashing=new node[size];

}

public void insertValue(String password,String name){

int ascii=0;

for (int i=0;i<password.length();i++){

char pasChar=password.charAt(i);

ascii=ascii+(int) pasChar;

}

node value=new node(password,name);

int ind=ascii%size;

if (hashing[ind]==null){

hashing[ind]=value;

}

else {

value.next=hashing[ind];

hashing[ind]=value;

}

}

public void deleteEntry(String password,String name){

int ascii=0;

for (int i=0;i<password.length();i++){

char pasChar=password.charAt(i);

ascii=ascii+(int) pasChar;

}

node value1=null;

int ind=ascii%size;

value1=hashing[ind];

node value2=null;

boolean flag=false;

while (value1.password!=password&&value1.name!=name&&value1.next!=null){

value2=value1;

value1=value1.next;

if (value1.password.equals(password)&&value1.name.equals(name)){

flag=true;

break;

}

}

if (flag){

value2.next=value1.next;

}

}

public void searchValue(String password,String name){

int ascii=0;

for (int i=0;i<password.length();i++){

char pasChar=password.charAt(i);

ascii=ascii+(int) pasChar;

}

int ind=ascii%size;

boolean flag=false;

while (hashing[ind]!=null){

if (hashing[ind].password.equals(password)&&hashing[ind].name.equals(name)){

flag=true;

break;

}

hashing[ind]=hashing[ind].next;

}

System.out.println(hashing[ind].name+" found at index: "+ind);

}

public void updatePassword(String password,String name){

int ascii=0;

for (int i=0;i<password.length();i++){

char pasChar=password.charAt(i);

ascii=ascii+(int) pasChar;

}

int ind=ascii%size;

node value1=null;

value1=hashing[ind];

if ((hashing[ind].password.equals(password))&&hashing[ind].name.equals(name)){

Scanner sc=new Scanner(System.in);

System.out.println("Enter the new password: ");

String password1=sc.next();

hashing[ind]=hashing[ind].next;

insertValue(password1,name);

}

else {

while ((hashing[ind].password!=password&&hashing[ind].name!=name)&&hashing[ind].next!=null){

value1=hashing[ind];

hashing[ind]=hashing[ind].next;

}

if (hashing[ind]!=null){

Scanner sc=new Scanner(System.in);

System.out.println("Enter the new password: ");

String password1=sc.next();

value1.next=hashing[ind].next.next;

insertValue(password1,name);

}

}

}

public void print(){

for (int i=0;i<hashing.length;i++){

System.out.println("Bucket "+i+" : ");

node value=hashing[i];

while (value!=null){

System.out.println("Name: "+value.name+" Password: "+value.password);

value=value.next;

}

}

}

public static void main(String[] args) {

Lab12Task5 hash=new Lab12Task5(5);

System.out.println("\*\*HashTable\*\*");

hash.insertValue("12345","Ahsan");

hash.insertValue("45434","Hassan");

hash.insertValue("54554","Hussain");

hash.insertValue("86432","Kashif");

hash.print();

System.out.println("\*\*After deleting an entry\*\*");

hash.deleteEntry("54554","Hussain");

hash.print();

System.out.println("\*\*Searching for the value\*\*");;

hash.searchValue("12345","Ahsan");

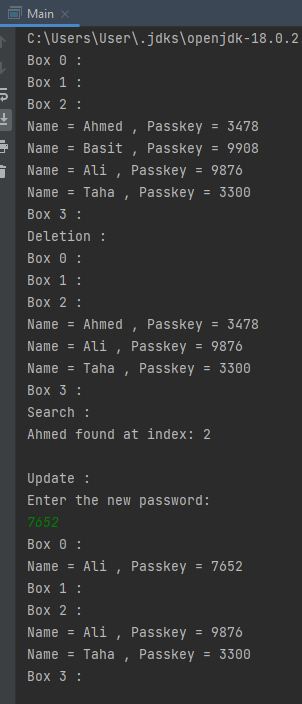
System.out.println();

System.out.println("\*\*Updating password\*\*");

hash.updatePassword("12345","Ahsan");

hash.print();

} }



**Question 06:**

public class Qno6\_Lab12 {

public static void main(String[] args) {

Rehashing r = new Rehashing();

r.Initial();

r.Insert(100);

r.Insert(101);

r.Insert(102);

r.Print();

r.Insert(103);

r.Print\_Hash();

}

}

class Rehashing{

int arr[] = new int[4];

int size=arr.length;

float hl = 0.75f;

int j=1;

int str[]= new int [8];

int k=0;

public void Initial(){

for(int i=0;i<size;i++){

arr[i]=-1;

}

}

public void Insert(int data){

int num = data%size;

float fv = j/size;

j++;

if(fv<hl){

if(arr[num]==-1){

arr[num]=data;

}

else{

for(int i=0;i<size;i++){

int key = (num+i)%size;

if(arr[key]==-1) {

arr[key]=data;

break;

}

}

}

}

else{

for(int i=0;i<str.length;i++){

str[i]=-1;

}

for(int i=0;i<size;i++){

if(arr[i]!=-1){

int n=arr[i]%str.length;

if(str[n]==-1){

str[n]=arr[i];

}

}

// k++;

}

int key = data%str.length;

if(str[key]==-1){

str[key]=data;

}

else{

for(int i=0;i<str.length;i++){

int j=(key+i)%str.length;

if(str[j]==-1) {

str[j]=data;

break;

}

}

}

}

}

public void Print\_Hash(){

for(int i=0;i<str.length;i++){

if(str[i]!=-1){

System.out.println("Index: "+i+" | Value: "+str[i]+" ");

}

}

}

public void Print(){

for(int i=0;i<size;i++){

if(arr[i]!=-1){

System.out.println("Index: "+i+" | Value: "+arr[i]+" ");

}

}

System.out.println();

}

} } }

