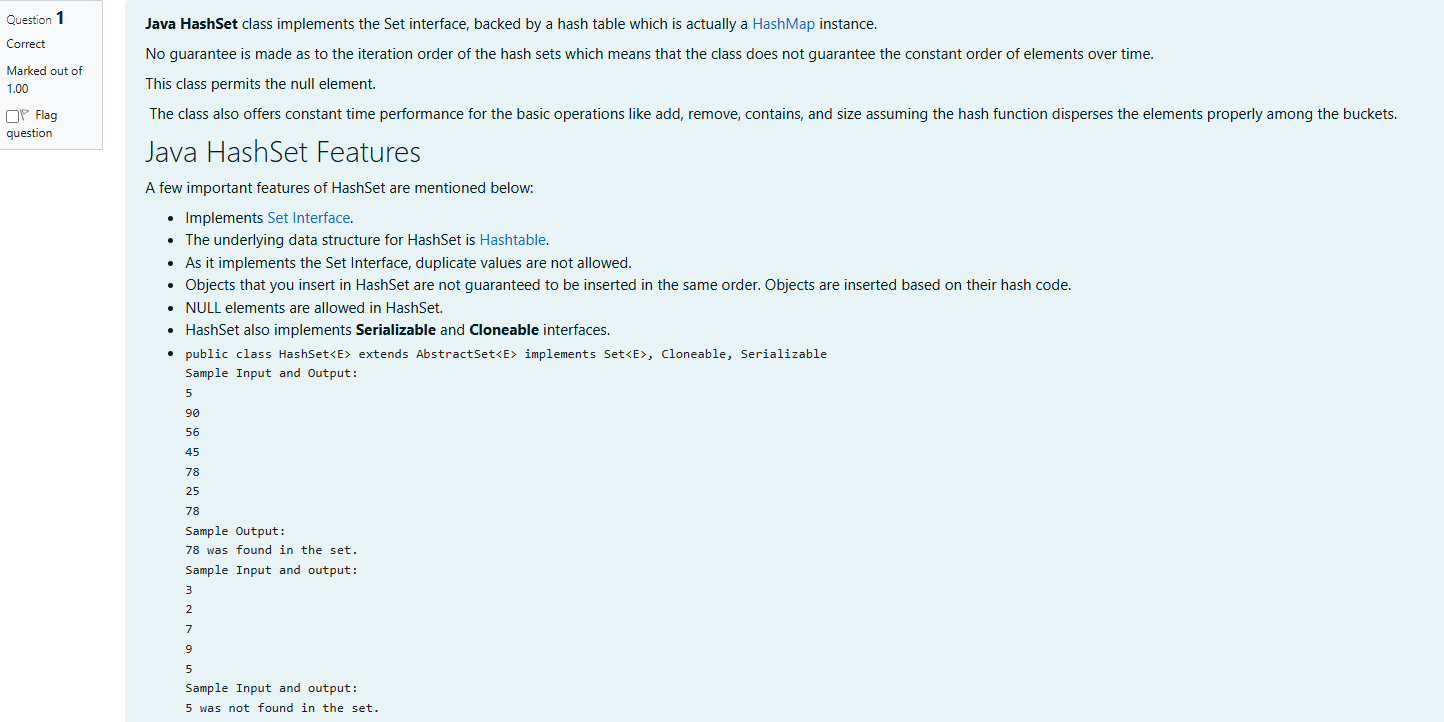
OBJECT ORIENTED PROGRAMMING USING JAVA

NAME : T.R.DIVYASREE

DEPT & SEC : CSE & B

ROLL NO : 230701083

WEEK : 11



import java.util.HashSet;

import java.util.Scanner;

class prog {

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

int n = sc.nextInt();

// Create a HashSet object called numbers

HashSet<Integer> numbers = new HashSet<>();

// Add values to the set

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

numbers.add(sc.nextInt());

}

int skey = sc.nextInt();

// Check if skey is in the set

if (numbers.contains(skey)) {

System.out.println(skey + " was found in the set.");

} else {

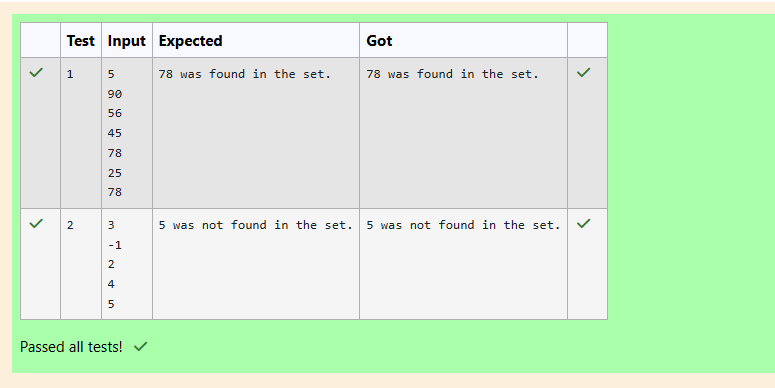
System.out.println(skey + " was not found in the set.");

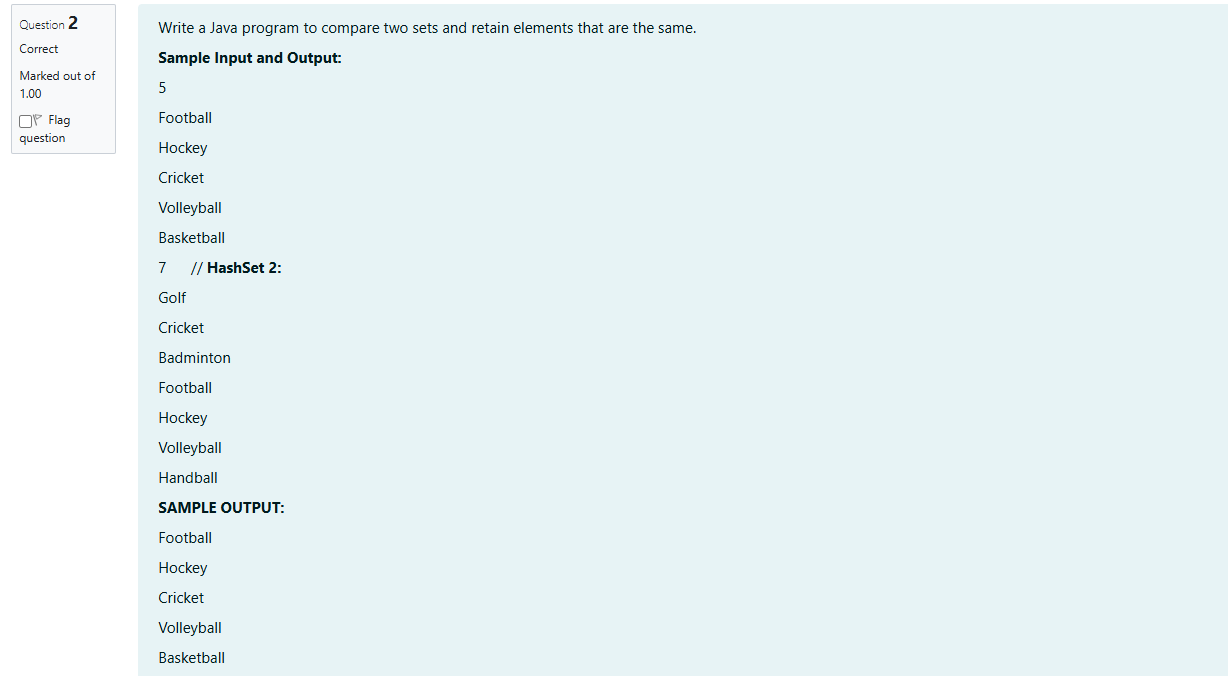
}

sc.close();

}

}





import java.util.HashSet;

import java.util.Scanner;

public class CompareSets {

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

// Input first set

int n1 = sc.nextInt();

sc.nextLine(); // Consume newline

HashSet<String> set1 = new HashSet<>();

for (int i = 0; i < n1; i++) {

set1.add(sc.nextLine());

}

// Input second set

int n2 = sc.nextInt();

sc.nextLine(); // Consume newline

HashSet<String> set2 = new HashSet<>();

for (int i = 0; i < n2; i++) {

set2.add(sc.nextLine());

}

// Retain only common elements in set1

set1.retainAll(set2);

for (String sport : set1) {

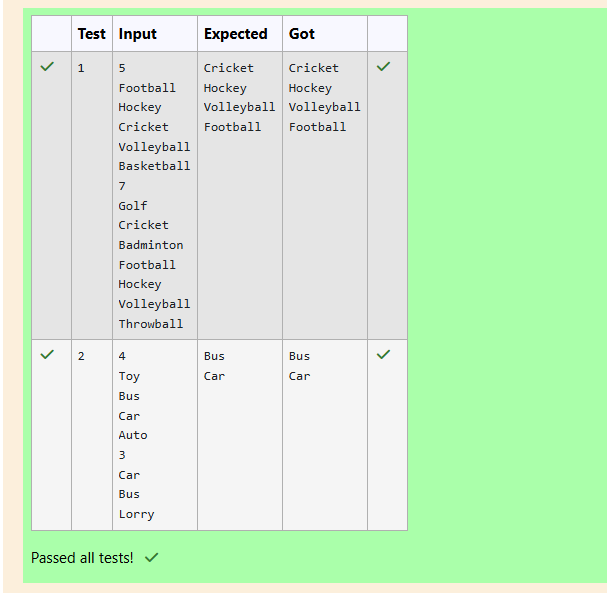
System.out.println(sport);

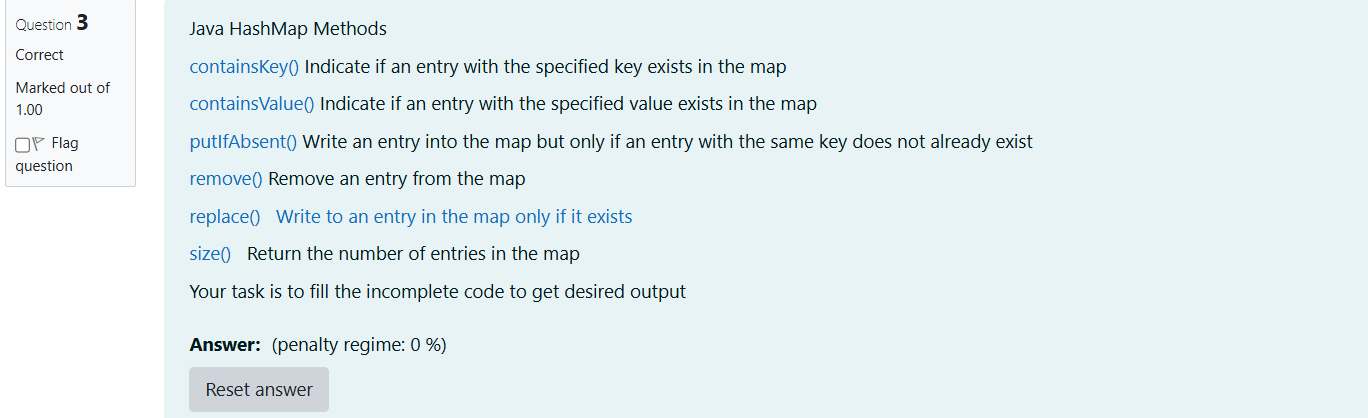
}

sc.close();

}

}





import java.util.HashMap;

import java.util.Map.Entry;

import java.util.Set;

import java.util.Scanner;

class prog {

public static void main(String[] args) {

// Creating HashMap with default initial capacity and load factor

HashMap<String, Integer> map = new HashMap<String, Integer>();

String name;

int num;

Scanner sc = new Scanner(System.in);

int n = sc.nextInt();

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

name = sc.next();

num = sc.nextInt();

map.put(name, num);

}

// Printing key-value pairs

Set<Entry<String, Integer>> entrySet = map.entrySet();

for (Entry<String, Integer> entry : entrySet) {

System.out.println(entry.getKey() + " : " + entry.getValue());

}

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

// Creating another HashMap

HashMap<String, Integer> anotherMap = new HashMap<String, Integer>();

// Inserting key-value pairs to anotherMap using put() method

anotherMap.put("SIX", 6);

anotherMap.put("SEVEN", 7);

// Inse

anotherMap.putAll(map); // Code to add all entries from 'map' to 'anotherMap'

// Printing key-value pairs of anotherMap

entrySet = anotherMap.entrySet();

for (Entry<String, Integer> entry : entrySet) {

System.out.println(entry.getKey() + " : " + entry.getValue());

}

// Adds key-value pair 'FIVE-5' only if it is not present in map

map.putIfAbsent("FIVE", 5);

// Retrieving a value associated with key 'TWO'

int value = map.getOrDefault("TWO", -1); // Using getOrDefault to avoid null if 'TWO' is not present

System.out.println(value);

// Checking whether key 'ONE' exists in map

System.out.println(map.containsKey("ONE"));

// Checking whether value '3' exists in map

System.out.println(map.containsValue(3));

// Retrieving the number of key-value pairs present in map

System.out.println(map.size());

sc.close();

}

}

