Data Structures and Algo in Java

Data structures sheet

import java.util.\*;

public class dataStructures{

public static void main(String[] args)

{

// return true if duplicates are present

// Input: [1, 2, 3, 4, 5, 2]

// Output: true

// return a string by accessing in index

// Input: add("Song A"), add("Song B"), get(1)

// Output: "Song B"

// to count the letters

// Input: "banana"

// Output: {b=1, a=3, n=2}

// appear more than once

// Input: [1, 2, 2, 3, 4, 1, 5]

// Output: [1, 2]

int arr0 [] = {1,2,3,4,5,2};

checkdubplicates(arr0);

String arr1 [] = {"Song A","Song B","Song C","Song D"};

int get = 3;

getsong(arr1,get);

String s = "banana";

count(s);

int arr2 [] = {1,2,2,3,4,1,5,5};

duplicates(arr2);

}

public static void checkdubplicates(int arr[])

{

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

for(int num : arr)

{

hashset.add(num);

}

if(arr.length == hashset.size())

{

System.out.println("There are no duplicates");

}

else

{

System.out.println("There are duplicates");

}

}

public static void getsong(String arr [], int get)

{

ArrayList<String> arrlist = new ArrayList<>();

for(String each : arr)

{

arrlist.add(each);

}

System.out.println(arrlist.get(get-1));

}

public static void count (String s)

{

//banana

int count =1;

HashMap<Character,Integer> hashmap = new HashMap<>();

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

{

char ch = s.charAt(i);

if(hashmap.containsKey(ch))

{

hashmap.put(ch,hashmap.get(ch)+1);

}

else

{

hashmap.put(ch,1);

}

}

System.out.println(hashmap);

}

public static void duplicates(int arr [])

{

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

Set<Integer> set = new HashSet<>();

int count = 1;

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

{

if(map.containsKey(arr[i]))

{

map.put(arr[i],count+1);

set.add(arr[i]);

}

else

{

map.put(arr[i],count);

}

}

System.out.println(set);

}

}