Roman to Integer

We store Roman character and its value in the hashmap and iterate over the sting and if the value is the less than the next value then subtract it otherwise add.

1. Valid Anagram: Anagram means same characters in the two strings

Check two strings length same otherwise return false

Store each character and its count into map for both the strings

Iterate over the string and and check the each count.

public class ValidAnagram {  
 public static void main(String[] args) {  
  
 String s1="ramudxcvbd";  
 String s2="ramucvbdvb12";  
  
 if(s1.length() !=s2.length()){  
 System.*out*.println("is not anagram");  
 return;  
 }  
  
 Map<Character,Integer> hm1=new HashMap<>();  
 Map<Character,Integer> hm2=new HashMap<>();  
  
 Map<Character, Long> collect = s1.chars().mapToObj(x -> (char) x).collect(Collectors.*groupingBy*(x -> x, Collectors.*counting*()));  
 Map<Character, Long> collect2 = s2.chars().mapToObj(x -> (char) x).collect(Collectors.*groupingBy*(x -> x, Collectors.*counting*()));  
  
 Set<Map.Entry<Character, Long>> entries = collect.entrySet();  
 boolean isAnagram=false;  
 for(int i=0;i<s1.length();i++){  
  
 if(collect.get(s1.charAt(i)) ==collect.get(s2.charAt(i)) ){  
 isAnagram=true;  
 }else{  
 isAnagram=false;  
 }  
 }  
  
 System.*out*.println(isAnagram+"isAnagram");  
 }  
}