Strings :

1. Is palindrome(string A )

Public int isPalindrome(String str){

String givenString = Str.replaceAll(“[a-zA-Z0-9],”);

StringBuilder builder = new StringBuilder(givenString);

String reversedString = builder.reverse().toString();

If(givenString .equalsIgnoreCase(reversedString))

Return 1;

Else

Return 0;

}

1. Vowel and consonant substrings

Input 1:

A = "aba"

Substrings of S are : [a, ab, aba, b, ba, a]Out of these only 'ab' and 'ba' satisfy the condition for special Substring. So the answer is 2.

Public int countsubstrings(String str){

Int mod=10000000007;

Int result=0;

Int vowelcount=0,consonantcount=0;

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

Char ch = str.charAt(i);

If(isVowel(ch)){

Vowelcount++;

}

Else{

Consonantcount++;

}

Result=(result+ (ch==’a’ || ch==’e’ || ch==’I’|| ch==’o’ || ch==’u’ ||

Ch==’A’ || ch==’E’ || ch==’I’ || ch==’O’ || ch==’U’ ) ? cosonantcount : vowelcount ))%mod;

}

Return result;

}

Public Boolean isVowel(char ch){

If( ch ==’a’ || ch==’e’ || ch=’I’ ||ch==’o’ || ch==’u’ || ch==’A’ || ch==’E’ || ch ==’I’ || ch==’O’ || ch==’U” )

Return true;

Else return false;

}

1. Remove consecutive characters

Input 1:

A = "aabcd"

B = 2

Input 2:

A = "aabbccd"

B = 2

**Example Output**

Output 1:

"bcd"

Output 2:

"d"

Public string remove(String str, int b) {

String resultString = “”;

Int i=0,j=1;

Int count=0;

While(j<str.length()){

If(str.charAt(i)==str.charAt(j)){

Count++;

}else{

If(str.charAt(i)!=str.charAt(j)){

If(count!=b){

resultString+=str.subString(i,j));

}

Count=1;

}

I=j;

}

J++;

}

If(count!=b){

resultString+=str.subString(I,j);

}

Return resultString;

}

1. Serialize

A = ['scaler', 'academy']

Input 2:

A = ['interviewbit']

**Example Output**

Output 1:

scaler6~academy7~

Output 2:

interviewbit12~

public String serialize(ArrayList<String> str){

String finalString=””;

For(String arrayString : str){

Int length = str.length();

finalString+=str+length+”~”;

}

Return finalString;

}

1. Deserialize

**Example Input**

Input 1:

A = 'scaler6~academy7~'

Input 2:

A = 'interviewbit12~'

**Example Output**

Output 1:

['scaler', 'academy']

Output 2:

['interviewbit']

Public ArrayList<String> deserialize ( String str){

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

String addToArray=””;

Int j=0;

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

Char ch= str.charAt(i);

If(ch==’~’ && i<str.length()){

J=i+1;

}

If(Character.isDigit(ch)){

addToArray+=str.substring(j,i);

if(!addToArray.matches(“?\\d+”)){

list.add(addToArray);

}

addToArray=””;

j=I;

}

}

Return list;

}

1. String and its Frequency

**Example Input**

Input 1:

abbhuabcfghh

Input 2:

a

**Example Output**

Ouput 1:

a2b3h3u1c1f1g1

Ouput 2:

a1

public String frequencyString(String input){

Map<Character,Integer> frequencyMap= new HashMap<>();

String resultString=””;

For(int i=0; i<input.length();i++){

If(frequencyMap.get(input.charAt(i))!= null){

frequencyMap.put(input.charAt(i),frequencyMap.get(input.charAt(i))+1);

}

Else{

frequencyMap.put(input.charAt(i),1);

}

}

For(int i=0; i<input.length();i++){

If(frequencyMap.get(input.charAt(i))!=0){

resultString+=input.charAt(i);

resultString+=frequencyMap.get(input.charAt(i));

frequencymap.put(input.charAt(i),0);

}

}

Return resultString;

}

1. Bulls And Cows Problem

**Example Input**

Input 1:

secret = "1807", guess = "7810"

Input 2:

secret = "1123", guess = "0111"

**Example Output**

Ouput 1:

"1A3B"

Ouput 2:

"1A1B"

Public String BullCow(String secret, String guess){

Int sec[] = new int[10];

Int gue[] = new int[10];

Int bull=0;

For(int i=0; i<secret.length();i++){

If(secret.charAt(i)==guess.charAt(i)){

Bull++;

}else{

Sec[secret.charAt(i)-‘0’]++;

Gue[secret.charAt(i)-‘0’]++;

}

}

Int cows=0;

For(int i=0;i<10;i++){

Cows+=Math.min(sec[i],gue[i]);

}

Return bull+”A”+cows+”B”;

}

1. Self permutation

**Example Input**

Input 1:

A = 'scaler'

B = 'relasc'

Input 2:

A = 'scaler'

B = 'interviewbit'

**Example Output**

Output 1:

1

Output 2:

0

Public int selfPermutation(String str1, String str2){

Int characters[] = new int[256];

If(str1.length()!=str2.length()){

Return 0;

}else{

For(int i=0; i<str1.length();i++){

Characters[str1.charAt(i)]++;

Characters[str2.charAt(i)]--;

}

For(int i=0; i<256;i++){

If(character[i]!=0) return 0;

}

Return 1;

}

}

1. Longest common prefix

**Example Input**

Input 1:

A = ["abcdefgh", "aefghijk", "abcefgh"]

Input 2:

A = ["abab", "ab", "abcd"];

**Example Output**

Output 1:

"a"

Output 2:

"ab"

Public string lcp(ArrayList<String> givenStrs){

If(givenStrs.length==1) return givenStrs[0];

String resultString = givenStrs[0];

For(int i=1; i<givenStrs.length();i++){

while(givenStrs.get(i).indexOf(resultString)!=0){

resultString = resultString.substring(0,resultString.length()-1);

if(resultString.isEmpty()) return “”;

}

}

Return resultString;

}

1. Count and Say

The count-and-say sequence is the sequence of integers beginning as follows:  1, 11, 21, 1211, 111221, ... 1 is read off as one 1 or 11. 11 is read off as two 1s or 21.

21 is read off as one 2, then one 1 or 1211.

Given an integer n, generate the nth sequence.

Note: The sequence of integers will be represented as a string.

**Example:**

if n = 2, the sequence is 11.

Public String countnSay(int n){

String ans=”1”;

For(int i=1; i<n;i++){

Ans = generate(ans);

}

Return ans;

}

String generate(String str){

String ans=””;

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

For(int i=0; str.length()+1; i++){

If(i==str.length() || i<str.length() && map.get(str.charAt(i))==null){

Ans+=map.get(str.charAt(i))+str.charAt(i);

Map.clear();

}

If(i==str.length())

Map.put(null,0);

If(map.get(str.charAt(i)!=null){

Map.put(str.charAt(i), map.get(str.charAt(i))+1);

}else{

If(i<str.length()) map.put(str.chatAt(i),0));

}

}

Return ans;

}

1. Amazing SubArrays

**Example Input**

ABEC

**Example Output**

6

**Example Explanation**

Amazing substrings of the given string are :  
1. A  
2. AB  
3. ABE  
4. ABEC  
5. E  
6. EC  
here the number of substrings is 6 and 6 % 10003 = 6.

Public int subarray(String str){

Int count=0;

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

Char ch = str.charAt(i);

If(isVowel(ch)) {

Count+=str.length()-I;

}

}

Return count;

}

Boolean isVowel ( char ch){

Return “aeiouAEIOU”.indexOf(ch)!=-1;

}