Java Program to find all the permutations of a string

import java.util.ArrayList;

import java.util.List;

public class StringPermutations {

// Method to generate all permutations of the string

public static void generatePermutations(String str, int index, List<String> result) {

// Base case: if we have considered all characters

if (index == str.length()) {

result.add(str);

return;

}

// Recursively generate permutations

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

// Swap the character at index with the character at i

str = swap(str, index, i);

// Recursively generate permutations of the remaining substring

generatePermutations(str, index + 1, result);

// Backtrack: undo the swap to explore other possibilities

str = swap(str, index, i);

}

}

// Method to swap characters at positions i and j

private static String swap(String str, int i, int j) {

char[] charArray = str.toCharArray();

char temp = charArray[i];

charArray[i] = charArray[j];

charArray[j] = temp;

return new String(charArray);

}

public static void main(String[] args) {

String input = "ABC"; // Sample input string

List<String> permutations = new ArrayList<>();

// Generate all permutations starting from index 0

generatePermutations(input, 0, permutations);

// Output the generated permutations

System.out.println("All permutations of the string are:");

for (String perm : permutations) {

System.out.println(perm);

}

}

}

Java Program to find the longest repeating sequence in a string

1. public class LongestRepeatingSequence {

2. //Checks for the largest common prefix

3. public static String lcp(String s, String t){

4. int n = Math.min(s.length(),t.length());

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

6. if(s.charAt(i) != t.charAt(i)){

7. return s.substring(0,i);

8. }

9. }

10. return s.substring(0,n);

11. }

12.

13. public static void main(String[] args) {

14. String str = "acbdfghybdf";

15. String lrs="";

16. int n = str.length();

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

18. for(int j = i+1; j < n; j++){

19. //Checks for the largest common factors in every substring

20. String x = lcp(str.substring(i,n),str.substring(j,n));

21. //If the current prefix is greater than previous one

22. //then it takes the current one as longest repeating sequence

23. if(x.length() > lrs.length()) lrs=x;

24. }

25. }

26. System.out.println("Longest repeating sequence: "+lrs);

27. }

28. }

Java Program to replace lower-case characters with upper-case and vice-versa

1. public **class** changeCase {
2. **public** **static** **void** main(String[] args) {
4. String str1="Great Power";
5. StringBuffer newStr=**new** StringBuffer(str1);
7. **for**(**int** i = 0; i < str1.length(); i++) {
9. //Checks for lower case character
10. **if**(Character.isLowerCase(str1.charAt(i))) {
11. //Convert it into upper case using toUpperCase() function
12. newStr.setCharAt(i, Character.toUpperCase(str1.charAt(i)));
13. }
14. //Checks for upper case character
15. **else** **if**(Character.isUpperCase(str1.charAt(i))) {
16. //Convert it into upper case using toLowerCase() function
17. newStr.setCharAt(i, Character.toLowerCase(str1.charAt(i)));
18. }
19. }
20. System.out.println("String after case conversion : " + newStr);
21. }
22. }