

Print all possible strings of length k that can be formed from a set of n characters

Given a set of characters and a positive integer k, print all possible strings of length k that can be formed from the given set.

Examples:

Input:

```
set[] = {'a', 'b'}, k = 3
```

Output:

```
aaa
aab
aba
abb
baa
bab
bba
bbb
```

Input:

```
set[] = {'a', 'b', 'c', 'd'}, k = 1
```

Output:

```
a
b
```

Google™ Custom Search



GeeksforGeeks



53,527 people like GeeksforGeeks.



Facebook

Interview Experiences

Advanced Data Structures

Dynamic Programming

Greedy Algorithms

Backtracking

Pattern Searching

Divide & Conquer

Mathematical Algorithms

Recursion

c
d

For a given set of size n , there will be n^k possible strings of length k . The idea is to start from an empty output string (we call it *prefix* in following code). One by one add all characters to *prefix*. For every character added, print all possible strings with current prefix by recursively calling for k equals to $k-1$.

Following is Java implementation for same.

```
// Java program to print all possible strings of length k
class PrintAllKLengthStrings {

    // Driver method to test below methods
    public static void main(String[] args) {
        System.out.println("First Test");
        char set1[] = {'a', 'b'};
        int k = 3;
        printAllKLength(set1, k);

        System.out.println("\nSecond Test");
        char set2[] = {'a', 'b', 'c', 'd'};
        k = 1;
        printAllKLength(set2, k);
    }

    // The method that prints all possible strings of length k. It is
    // mainly a wrapper over recursive function printAllKLengthRec()
    static void printAllKLength(char set[], int k) {
        int n = set.length;
        printAllKLengthRec(set, "", n, k);
    }

    // The main recursive method to print all possible strings of length k
    static void printAllKLengthRec(char set[], String prefix, int n, int k) {

        // Base case: k is 0, print prefix
        if (k == 0) {
            System.out.println(prefix);
            return;
        }

        // One by one add all characters from set and recursively
        // call for k equals to k-1
        for (int i = 0; i < n; ++i) {
```



Popular Posts

All permutations of a given string

Memory Layout of C Programs

Understanding "extern" keyword in C

Median of two sorted arrays

Tree traversal without recursion and without stack!

Structure Member Alignment, Padding and

Data Packing

Intersection point of two Linked Lists

Lowest Common Ancestor in a BST.

Check if a binary tree is BST or not

Sorted Linked List to Balanced BST

```
// Next character of input added
String newPrefix = prefix + set[i];

// k is decreased, because we have added a new character
printAllKLengthRec(set, newPrefix, n, k - 1);
    }
}
```

Output:

First Test

aaa
aab
aba
abb
baa
bab
bba
bbb

Second Test

a
b
c
d

The above solution is mainly generalization of [this post](#).

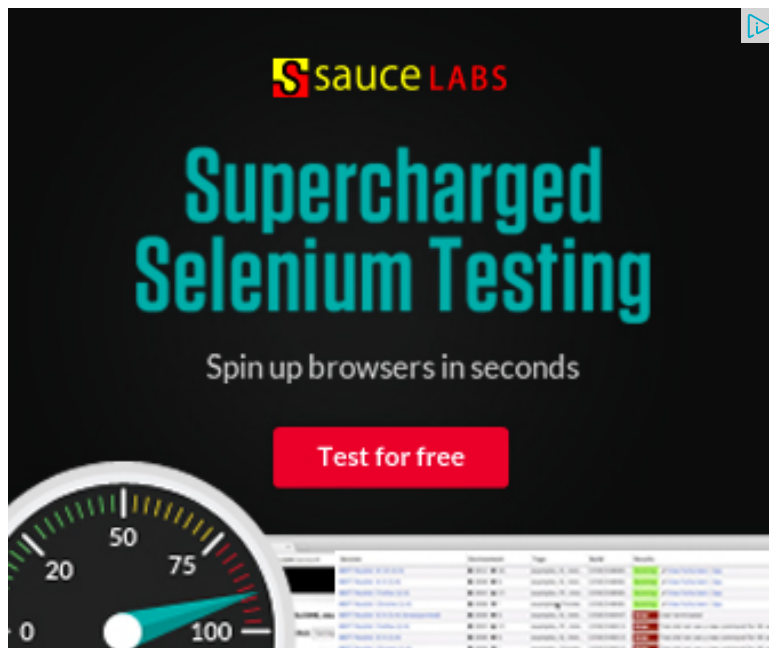
This article is contributed by **Abhinav Ramana**. Please write comments if you find anything incorrect, or you want to share more information about the topic discussed above

Deploy Early. Deploy Often.

DevOps from
Rackspace:

Automation

FIND OUT HOW ►



705



Subscribe

Recent Comments

Abhi You live US or India?

Google (Mountain View) interview · 35 minutes ago

Aman Hi, Why arent we checking for conditions...

Write a C program to Delete a Tree. · 1 hour ago

kzs please provide solution for the problem...

Backtracking | Set 2 (Rat in a Maze) · 1 hour ago

Sanjay Agarwal bool

tree::Root_to_leaf_path_given_sum(tree...

Root to leaf path sum equal to a given number · 1 hour ago

GOPI GOPINATH @admin Highlight this sentence "We can easily...

Count trailing zeroes in factorial of a number · 1 hour ago

newCoder3006 If the array contains negative numbers also. We...

Find subarray with given sum · 2 hours ago

Related Topics:

- [Backtracking | Set 8 \(Solving Cryptarithmic Puzzles\)](#)
- [Tail Recursion](#)
- [Find if two rectangles overlap](#)
- [Analysis of Algorithm | Set 4 \(Solving Recurrences\)](#)
- [Print all possible paths from top left to bottom right of a mXn matrix](#)
- [Generate all unique partitions of an integer](#)
- [Russian Peasant Multiplication](#)
- [Closest Pair of Points | O\(nlogn\) Implementation](#)



25



Tweet

3



2


Writing code in comment? Please use [ideone.com](#) and share the link here.

AdChoices 

► [String Function](#)

► [String Java](#)


► [C++ Begin String](#)

AdChoices 

► [C String](#)

► [C# String Split](#)

► [String File](#)

AdChoices 

► [C# String Split](#)

► [String Class](#)

► [Just String com](#)

@geeksforgeeks, **Some rights reserved**

Contact Us!

Powered by **WordPress** & **MooTools**, customized by geeksforgeeks team