

(<http://www.sanfoundry.com>)



Questions & Answers

- C Interview Questions (<http://www.sanfoundry.com/c-interview-questions-answers/>)
- C++ Questions (<http://www.sanfoundry.com/cplusplus-interview-questions-answers/>)
- Linux MCQs (<http://www.sanfoundry.com/technical-interview-questions/>)
- C# Quiz (<http://www.sanfoundry.com/csharp-questions-answers/>)
- Java MCQs (<http://www.sanfoundry.com/java-questions-answers-freshers-experienced/>)
- JavaScript MCQs (<http://www.sanfoundry.com/javascript-questions-answers/>)
- SAN Questions (<http://www.sanfoundry.com/san-storage-mcqs-freshers-experienced/>)
- PHP Questions (<http://www.sanfoundry.com/php-questions-answers/>)
- Python Quiz (<http://www.sanfoundry.com/1000-python-questions-answers/>)

Computer Science Questions

- Operating System Quiz (<http://www.sanfoundry.com/operating-system-questions-answers/>)
- Computer Architecture MCQs (<http://www.sanfoundry.com/computer-organization-architecture-questions-answers/>)
- Software Architecture MCQs (<http://www.sanfoundry.com/software-architecture-design-questions-answers/>)
- Software Engineering MCQs (<http://www.sanfoundry.com/software-engineering-questions-answers/>)
- Artificial Intelligence MCQs (<http://www.sanfoundry.com/artificial-intelligence-questions-answers/>)
- LISP Programming MCQs (<http://www.sanfoundry.com/lisp-programming-questions-answers/>)
- Database Management MCQs (<http://www.sanfoundry.com/database-management-system-questions-answers/>)
- Computer Network MCQs (<http://www.sanfoundry.com/computer-network-questions-answers/>)
- Microprocessor MCQs (<http://www.sanfoundry.com/microprocessors-questions-answers/>)

C Programming Examples

- Simple C Programs (<http://www.sanfoundry.com/simple-c-programs/>)
- C - Arrays (<http://www.sanfoundry.com/c-programming-examples-arrays/>)
- C - Matrix (<http://www.sanfoundry.com/c-programming-examples-matrix/>)
- C - Strings (<http://www.sanfoundry.com/c-programming-examples-strings/>)
- C - Bitwise Operations (<http://www.sanfoundry.com/c-programming-examples-bitwise-operations/>)
- C - Linked Lists (<http://www.sanfoundry.com/c-programming-examples-linked-list/>)
- C - Stacks & Queues (<http://www.sanfoundry.com/c-programming-examples-stacks/>)
- C - Searching & Sorting (<http://www.sanfoundry.com/c-programming-examples-searching-sorting/>)
- C - Trees (<http://www.sanfoundry.com/c-programming-examples-on-trees/>)
- C - Strings (<http://www.sanfoundry.com/c-programming-examples-strings/>)
- C - File Handling (<http://www.sanfoundry.com/c-programming-examples-file-handling/>)
- C - Mathematical Functions (<http://www.sanfoundry.com/c-programming-examples-mathematical-functions/>)
- C - Puzzles & Games (<http://www.sanfoundry.com/c-programming-examples-on-puzzles-games/>)
- C Programs - Recursion (<http://www.sanfoundry.com/c-programming-examples-recursion/>)
- C Programs - No Recursion (<http://www.sanfoundry.com/c-programming-examples-without-using-recursion/>)

Java Algorithms

- Java - Numerical Problems (<http://www.sanfoundry.com/java-programming-examples-numerical-problems-algorithms/>)
- Java - Combinatorial Problems (<http://www.sanfoundry.com/java-programming-examples-combinatorial-problems->

algorithms/)

Java - Graph Problems (<http://www.sanfoundry.com/java-programming-examples-graph-problems-algorithms/>)

Java - Hard Graph Problems (<http://www.sanfoundry.com/java-programming-examples-hard-graph-problems-algorithms/>)

Java - Computation Geometry (<http://www.sanfoundry.com/java-programming-examples-computational-geometry-problems-algorithms/>)

Java - Sets & Strings (<http://www.sanfoundry.com/java-programming-examples-set-string-problems-algorithms/>)

Java - Data-Structures (<http://www.sanfoundry.com/java-programming-examples-data-structures/>)

Java - Collection API Problems (<http://www.sanfoundry.com/java-programming-examples-collection-api/>)

C++ Algorithms

C++ - Numerical Problems (<http://www.sanfoundry.com/cpp-programming-examples-numerical-problems-algorithms/>)

C++ - Combinatorial Problems (<http://www.sanfoundry.com/cpp-programming-examples-combinatorial-problems-algorithms/>)

C++ - Graph Problems (<http://www.sanfoundry.com/cpp-programming-examples-graph-problems-algorithms/>)

C++ - Hard Graph Problems (<http://www.sanfoundry.com/cpp-programming-examples-hard-graph-problems-algorithms/>)

C++ - Computation Geometry (<http://www.sanfoundry.com/cpp-programming-examples-computational-geometry-problems-algorithms/>)

C++ - Sets & Strings (<http://www.sanfoundry.com/cpp-programming-examples-set-string-problems-algorithms/>)

C++ - Data-Structures (<http://www.sanfoundry.com/cpp-programming-examples-data-structures/>)

C++ - STL Library (<http://www.sanfoundry.com/cpp-programming-examples-stl/>)

C Algorithms

C - Numerical Problems (<http://www.sanfoundry.com/c-programming-examples-numerical-problems-algorithms/>)

C - Combinatorial Problems (<http://www.sanfoundry.com/c-programming-examples-combinatorial-problems-algorithms/>)

C - Graph Problems (<http://www.sanfoundry.com/c-programming-examples-graph-problems-algorithms/>)

C - Hard Graph Problems (<http://www.sanfoundry.com/c-programming-examples-hard-graph-problems-algorithms/>)

C - Computation Geometry (<http://www.sanfoundry.com/c-programming-examples-computational-geometry-problems-algorithms/>)

C - Sets & Strings (<http://www.sanfoundry.com/c-programming-examples-set-string-problems-algorithms/>)

C - Data-Structures (<http://www.sanfoundry.com/c-programming-examples-data-structures/>)

Java Program to Implement Trie

This is a [Java Program](#) to implement Trie. A trie is an ordered [tree data structure](#) that is used to store a dynamic set or associative array where the keys are usually strings. Unlike a binary search tree, no node in the tree stores the key associated with that node, instead, its position in the tree defines the key with which it is associated. All the descendants of a node have a common prefix of the string associated with that node, and the root is associated with the empty string. Values are normally not associated with every node, only with leaves and some inner nodes that correspond to keys of interest.

Here is the source code of the Java program to implement Trie. The Java program is successfully compiled and run on a Windows system. The [program](#) output is also shown below.

```
1. /*
2.  * Java Program to Implement Trie
3.  */
4.
5. import java.util.*;
```

```
6.
7. class TrieNode
8. {
9.     char content;
10.    boolean isEnd;
11.    int count;
12.    LinkedList<TrieNode> childList;
13.
14.    /* Constructor */
15.    public TrieNode(char c)
16.    {
17.        childList = new LinkedList<TrieNode>();
18.        isEnd = false;
19.        content = c;
20.        count = 0;
21.    }
22.    public TrieNode subNode(char c)
23.    {
24.        if (childList != null)
25.            for (TrieNode eachChild : childList)
26.                if (eachChild.content == c)
27.                    return eachChild;
28.        return null;
29.    }
30. }
31.
32. class Trie
33. {
34.    private TrieNode root;
35.
36.    /* Constructor */
37.    public Trie()
38.    {
39.        root = new TrieNode(' ');
40.    }
41.    /* Function to insert word */
42.    public void insert(String word)
43.    {
44.        if (search(word) == true)
45.            return;
46.        TrieNode current = root;
47.        for (char ch : word.toCharArray() )
48.        {
49.            TrieNode child = current.subNode(ch);
50.            if (child != null)
51.                current = child;
52.            else
53.            {
54.                current.childList.add(new TrieNode(ch));
55.                current = current.subNode(ch);
56.            }
57.            current.count++;
```

```

58.     }
59.     current.isEnd = true;
60. }
61. /* Function to search for word */
62. public boolean search(String word)
63. {
64.     TrieNode current = root;
65.     for (char ch : word.toCharArray() )
66.     {
67.         if (current.subNode(ch) == null)
68.             return false;
69.         else
70.             current = current.subNode(ch);
71.     }
72.     if (current.isEnd == true)
73.         return true;
74.     return false;
75. }
76. /* Function to remove a word */
77. public void remove(String word)
78. {
79.     if (search(word) == false)
80.     {
81.         System.out.println(word + " does not exist in trie\n");
82.         return;
83.     }
84.     TrieNode current = root;
85.     for (char ch : word.toCharArray())
86.     {
87.         TrieNode child = current.subNode(ch);
88.         if (child.count == 1)
89.         {
90.             current.childList.remove(child);
91.             return;
92.         }
93.         else
94.         {
95.             child.count--;
96.             current = child;
97.         }
98.     }
99.     current.isEnd = false;
100. }
101. }
102.
103. /* Class Trie Test */
104. public class TrieTest
105. {
106.     public static void main(String[] args)
107.     {
108.         Scanner scan = new Scanner(System.in);
109.         /* Creating object of AATree */

```

```

110.     Trie t = new Trie();
111.     System.out.println("Trie Test\n");
112.     char ch;
113.     /* Perform tree operations */
114.     do
115.     {
116.         System.out.println("\nTrie Operations\n");
117.         System.out.println("1. insert ");
118.         System.out.println("2. delete");
119.         System.out.println("3. search");
120.
121.         int choice = scan.nextInt();
122.         switch (choice)
123.         {
124.             case 1 :
125.                 System.out.println("Enter string element to insert");
126.                 t.insert( scan.next() );
127.                 break;
128.             case 2 :
129.                 System.out.println("Enter string element to delete");
130.                 try
131.                 {
132.                     t.remove( scan.next() );
133.                 }
134.                 catch (Exception e)
135.                 {
136.                     System.out.println(e.getMessage()+" not found ");
137.                 }
138.                 break;
139.             case 3 :
140.                 System.out.println("Enter string element to search");
141.                 System.out.println("Search result : "+ t.search( scan.next() ));
142.                 break;
143.             default :
144.                 System.out.println("Wrong Entry \n ");
145.                 break;
146.         }
147.
148.         System.out.println("\nDo you want to continue (Type y or n) \n");
149.         ch = scan.next().charAt(0);
150.     } while (ch == 'Y' || ch == 'y');
151. }
152. }

```


Trie Test

Trie Operations

1. insert

2. delete

3. search

1

Enter string element to insert

trie

Do you want to **continue** (Type y or n)

y

Trie Operations

1. insert

2. delete

3. search

1

Enter string element to insert

tree

Do you want to **continue** (Type y or n)

y

Trie Operations

1. insert

2. delete

3. search

1

Enter string element to insert

branch

Do you want to **continue** (Type y or n)

y

Trie Operations

1. insert

2. delete

3. search

1

Enter string element to insert

beach

Do you want to **continue** (Type y or n)

y

Trie Operations

1. insert
2. delete
3. search

3

Enter string element to search

bean

Search result : **false**

Do you want to **continue** (Type y or n)

y

Trie Operations

1. insert
2. delete
3. search

3

Enter string element to search

beach

Search result : **true**

Do you want to **continue** (Type y or n)

y

Trie Operations

1. insert
2. delete
3. search

2

Enter string element to delete

bean

bean does not exist in trie

Do you want to **continue** (Type y or n)

y

Trie Operations

1. insert
2. delete
3. search

2

Enter string element to delete
beach

Do you want to **continue** (Type y or n)

y

Trie Operations

1. insert
 2. delete
 3. search
- 3

Enter string element to search

beach

Search result : **false**

Do you want to **continue** (Type y or n)

y

Trie Operations

1. insert
 2. delete
 3. search
- 2

Enter string element to delete

tree

Do you want to **continue** (Type y or n)

y

Trie Operations

1. insert
 2. delete
 3. search
- 3

Enter string element to search

tree

Search result : **false**

Do you want to **continue** (Type y or n)

y

Trie Operations

1. insert
2. delete

3. search

3

Enter string element to search

trie

Search result : **true**

Do you want to **continue** (Type y or n)

y

Trie Operations

1. insert

2. delete

3. search

2

Enter string element to delete

trie

Do you want to **continue** (Type y or n)

y

Trie Operations

1. insert

2. delete

3. search

3

Enter string element to search

trie

Search result : **false**

Do you want to **continue** (Type y or n)

n

Sanfoundry Global Education & Learning Series – 1000 Java Programs.

If you wish to look at all Java Programming examples, go to Java Programs (<http://www.sanfoundry.com/>).

If you liked this Java Program, kindly share, recommend or like below!




(<http://twitter.com/share>)

advertisements

Deep Dive @ Sanfoundry:

1. **Java Programming Examples on Set & String Problems & Algorithms** (<http://www.sanfoundry.com/java-programming-examples-set-string-problems-algorithms/>)
2. **100+ Java Android Programming Examples** (<http://www.sanfoundry.com/java-android-programming-examples/>)
3. **C Programming Examples on Arrays** (<http://www.sanfoundry.com/c-programming-examples-arrays/>)
4. **Java Programming Examples on Data-Structures** (<http://www.sanfoundry.com/java-programming-examples-data-structures/>)
5. **Java Programming Examples on Combinatorial Problems & Algorithms** (<http://www.sanfoundry.com/java-programming-examples-combinatorial-problems-algorithms/>)
6. **C Programming Examples on Strings** (<http://www.sanfoundry.com/c-programming-examples-strings/>)
7. **C# Programming Examples on Data Structures** (<http://www.sanfoundry.com/csharp-programming-examples-on-datastructures/>)
8. **Java Programming Examples on Graph Problems & Algorithms** (<http://www.sanfoundry.com/java-programming-examples-graph-problems-algorithms/>)

You May Like

Sponsored Links by Taboola 

9 Things to Keep in Mind Before Buying a Property in India

Bajaj Finserv

One Simple Way to Own a Superbike

MyUniverse

Calculate Now: How You Can Save Lakhs On Home Loan EMI

Big Decisions

5 Ways to Increase Your Annual Income

Scripbox

Emotional meeting between Indrani and daughter Vidhie

MSN

Why Hong Kong is the Most Exciting Travel Destination For Family?

Hong Kong Tourism Board

Manish Bhojasia (<http://www.sanfoundry.com/about/>), a technology veteran with 19+ years @ Cisco & Wipro, is Founder and CTO at Sanfoundry. He is Linux Kernel Developer and SAN Architect and is passionate about competency developments in these areas. He lives in Bangalore and delivers focused training sessions to IT professionals in Linux Kernel, Linux Debugging, Linux Device Drivers, Linux Networking, Linux Storage & Cluster Administration, Advanced C Programming, SAN Storage Technologies, SCSI Internals and Storage Protocols such as iSCSI & Fiber Channel. Stay connected with us below:

Google+ (<https://plus.google.com/104408026570656234343/posts>) | Facebook (<http://www.facebook.com/sanfoundry>) | Twitter (<http://www.twitter.com/sanfoundry>) | LinkedIn (<https://www.linkedin.com/company/sanfoundry>)

Subscribe Sanfoundry Newsletters & Posts

Name

Email Address

Subscribe

Best Careers

Developer Tracks (<http://www.sanfoundry.com/salary-50l/>)

SAN Developer (<http://www.sanfoundry.com/san-storage-developer-training-courses/>)

Linux Kernel Developer (<http://www.sanfoundry.com/linux-kernel-developer-training-courses-jobs/>)

Linux Driver Developer (<http://www.sanfoundry.com/linux-device-driver-developer-training/>)

Linux Network Developer (<http://www.sanfoundry.com/linux-network-developer-training/>)

Live Training Photos (<http://www.sanfoundry.com/sanfoundry-classes/>)

Mentoring (<http://www.sanfoundry.com/professional-mentoring-coaching-career-guidance-cto/>)

Software Productivity (<http://www.sanfoundry.com/programming-discipline-and-software/>)

GDB Assignment (<http://www.sanfoundry.com/gdb-example-tutorial/>)

Best Training

SAN I - Technology (<http://www.sanfoundry.com/san-storage-area-networks-training/>)

SAN II - Admin (<http://www.sanfoundry.com/san-administration-training-course/>)

Linux Fundamentals (<http://www.sanfoundry.com/linux-administration-training/>)

Advanced C Training (<http://www.sanfoundry.com/advanced-c-programming-training/>)

Linux-C Debugging (<http://www.sanfoundry.com/training-on-linux-debugging-techniques/>)

System Programming (<http://www.sanfoundry.com/training-on-linux-internals-systems/>)

Network Programming (<http://www.sanfoundry.com/training-socket-network-programming/>)

Linux Threads (<http://www.sanfoundry.com/training-multithreaded-parallel/>)

Kernel Programming (<http://www.sanfoundry.com/linux-kernel-internals-training/>)

Kernel Debugging (<http://www.sanfoundry.com/linux-kernel-debugging-training/>)

Linux Device Drivers (<http://www.sanfoundry.com/training-on-linux-device-drivers/>)

From The Web

9 Things to Keep in Mind Before Buying a Property in India

Bajaj Finserv

One Simple Way to Own a Superbike


MyUniverse

Calculate Now: How You Can Save Lakhs On Home Loan EMI

Big Decisions

5 Ways to Increase Your Annual Income

Scripbox

Sponsored Links by Taboola 

advertisements

Sanfoundry is **No. 1** choice for Deep Hands-ON Trainings in **SAN, Linux & C, Kernel Programming**. Our Founder has trained employees of almost all Top Companies in India such as VMware, Citrix, Oracle, Motorola, Ericsson, Aricent, HP, Intuit, Microsoft, Cisco, SAP Labs, Siemens, Symantec, Redhat, Chelsio, Cavium, ST-Micro, Samsung, LG-Soft, Wipro, TCS, HCL, IBM, Accenture, HSBC, Mphasis, Tata-Elxsi, Tata VSNL, Mindtree, Cognizant and Startups.

advertisements

