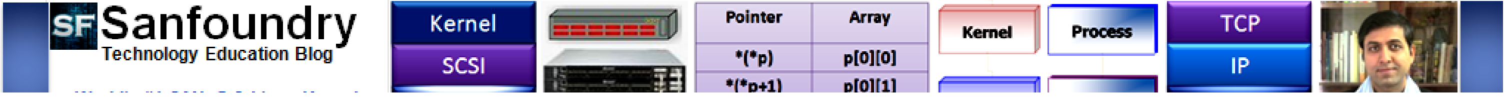

(http://www.sanfoundry.com)

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.

```
/*
 *  Java Program to Implement Trie
 */

import java.util.*;
```

```

class TrieNode
{
    char content;
    boolean isEnd;
    int count;
    LinkedList<TrieNode> childList;

    /* Constructor */
    public TrieNode(char c)
    {
        childList = new LinkedList<TrieNode>();
        isEnd = false;
        content = c;
        count = 0;
    }
    public TrieNode subNode(char c)
    {
        if (childList != null)
            for (TrieNode eachChild : childList)
                if (eachChild.content == c)
                    return eachChild;
        return null;
    }
}

class Trie
{
    private TrieNode root;

     /* Constructor */
    public Trie()
    {
        root = new TrieNode(' ');
    }
     /* Function to insert word */
    public void insert(String word)
    {
        if (search(word) == true)
            return;
        TrieNode current = root;
        for (char ch : word.toCharArray() )
        {
            TrieNode child = current.subNode(ch);
            if (child != null)
                current = child;
            else
            {
                 current.childList.add(new TrieNode(ch));
                 current = current.subNode(ch);
            }
            current.count++;
```

```java
        }
        current.isEnd = true;
    }
    /* Function to search for word */
    public boolean search(String word)
    {
        TrieNode current = root;
        for (char ch : word.toCharArray() )
        {
            if (current.subNode(ch) == null)
                return false;
            else
                current = current.subNode(ch);
        }
        if (current.isEnd == true)
            return true;
        return false;
    }
    /* Function to remove a word */
    public void remove(String word)
    {
        if (search(word) == false)
        {
            System.out.println(word +" does not exist in trie\n");
            return;
        }
        TrieNode current = root;
        for (char ch : word.toCharArray())
        {
            TrieNode child = current.subNode(ch);
            if (child.count == 1)
            {
                current.childList.remove(child);
                return;
            }
            else
            {
                child.count--;
                current = child;
            }
        }
        current.isEnd = false;
    }
}

/* Class Trie Test */
public class TrieTest
{
    public static void main(String[] args)
    {
        Scanner scan = new Scanner(System.in);
        /* Creating object of AATree */
    Trie t = new Trie();
    System.out.println("Trie Test\n");
    char ch;
    /*  Perform tree operations  */
    do
    {
      System.out.println("\nTrie Operations\n");
      System.out.println("1. insert ");
      System.out.println("2. delete");
      System.out.println("3. search");

      int choice = scan.nextInt();
      switch (choice)
      {
      case 1 :
        System.out.println("Enter string element to insert");
        t.insert( scan.next() );
        break;
      case 2 :
        System.out.println("Enter string element to delete");
        try
        {
          t.remove( scan.next() );
        }
        catch (Exception e)
        {
          System.out.println(e.getMessage()+" not found ");
        }
        break;
      case 3 :
        System.out.println("Enter string element to search");
        System.out.println("Search result : "+ t.search( scan.next() ));
        break;
      default :
        System.out.println("Wrong Entry \n ");
        break;
      }

      System.out.println("\nDo you want to continue (Type y or n) \n");
      ch = scan.next().charAt(0);
    } while (ch == 'Y'|| ch == 'y');
  }
}
```

```
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!

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-programing-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/)**

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