

Search:  [HOME](#) | [ARTICLES](#) | [FORUM](#) | [INTERVIEW FAQ](#) | [ACTIVITIES](#) | [NEWS](#) | [VIDEOS](#) | [POLL](#) | [LINKS](#) | [PEOPLE](#) | [GROUPS](#) |

## C++ Programming Articles

[Submit Article](#)[Home](#) » [Articles](#) » [C++ Programming](#) » [Parsing](#)[RSS Feeds](#)

# PROGRAM TO IMPLEMENT RECURSIVE DESCENT PARSER

**Posted By:** [Adalberto Fischer](#) **Category:** [C++ Programming](#) **Views:** 11483

## PROGRAM TO IMPLEMENT RECURSIVE DESCENT PARSER.

### Code for PROGRAM TO IMPLEMENT RECURSIVE DESCENT PARSER in C++ Programming

```
#include <iostream.h>
#include <stdio.h>
#include <conio.h>
#include <string.h>
#include <stdlib.h>

typedef struct treenode
{
    char info;
    treenode *left;
    treenode *right;
}node;

node * proc_e(char input[],int &ssm);
node * proc_t(char input[],int &ssm);
node * proc_v(char input[],int &ssm);
void traversal(node *temp);

void main()
{
    char input[20];
    int ssm=0;
    node *root;
    clrscr();

    cout<<endl<<"ENTER THE STRING TO BE PARSED:";
    cin>>input;

    root = proc_e(input,ssm);
    cout<<endl<<"CONTENTS OF THE PARSE TREE:";
    traversal(root);

    getch();
}

void traversal(node *temp)
{
    if(temp != NULL)
    {
        traversal(temp->left);
        cout<<temp->info;
        traversal(temp->right);
    }
}

//PROCEDURE FOR NT E:
node * proc_e(char input[],int &ssm)
{
    node *a,*b;

    a=proc_t(input,ssm);

    while(input[ssm]=='+' )
    {
        ssm++;
```

[C++ Programming](#)  
[View All](#)  
[Homework Help](#)  
[Data File Structure](#)  
[Computer Graphics](#)  
[Projects](#)  
[Beginners](#)  
[Object Oriented Progra...](#)  
[Algorithms](#)  
[Miscellaneous Problems](#)  
[Numerical Analysis](#)  
[Mathematics Program](#)  
[Mouse Programming](#)  
[Parsing](#)  
[Scanner](#)  
[Interview FAQ](#)[Assembly Language](#)[Artificial Intelligence](#)[C Programming](#)[Visual C++](#)[OOAD](#)[Cobol](#)[Java](#)[SQL Server](#)[Asp.net MVC](#)[Rest and WCF Services](#)[Entity Framework](#)[Knockout.js](#)[Unix / Linux / Ubuntu](#)[Networking](#)[OOPs Concept](#)[HTML](#)[Dos](#)[SQL](#)[System Analysis & Design](#)[Gadgets](#)[Internet](#)[CSS](#)[Javascript](#)[.Net Framework](#)[Asp.net](#)[C#](#)[VB.Net](#)[Python](#)[Perl](#)[Oracle](#)[Software Engineering](#)[RDBMS Terms](#)

```
b=proc_t(input,ssm);

node *temp;
temp=new node; //(node*)malloc(sizeof(node));
temp->info='+';
temp->left=a;
temp->right=b;

a=temp;

}

return a;
}

//PROCEDURE FOR NT T:

node * proc_t(char input[],int &ssm)
{
    node *a,*b;

    a=proc_v(input,ssm);
    ssm = ssm + 1;

    while(input[ssm] == '*')
    {
        ssm++;

        b=proc_v(input,ssm);
        node *temp;

        temp=new node;//(node *)malloc(sizeof(node));
        temp->info='*';
        temp->left=a;
        temp->right=b;

        a=temp;
        ssm = ssm +1;
    }

    return a;
}

//PROCEDURE FOR NT V:

node * proc_v(char input[],int &ssm)
{
    node *a;

    if(input[ssm]=='I')
    {
        node *temp;
        temp=new node; //(node*)malloc(sizeof(node));
        temp->info='I';
        temp->left=NULL;
        temp->right=NULL;
        return temp;
    }
    else
    {
        cout<<endl<<"ERROR. INVALID SYMBOL "<<input[ssm];
        getch();
        exit(0);
    }
}
```

Share:

**Previous Post:**

[To parse a string using Operator Precedence parsing](#)

**Next Post:**

[To parse a string using Recursive-Descent parser](#)

Didn't find what you were looking for? Find more on **PROGRAM TO IMPLEMENT RECURSIVE DESCENT PARSER** Or get **search suggestion and latest updates**.

Adalberto Fischer author of PROGRAM TO IMPLEMENT RECURSIVE DESCENT PARSER is from Frankfurt, [Germany](#).

[AJAX Framework](#)[Design Pattern](#)[UML](#)[WPF](#)[WCF](#)[SEO](#)[PowerShell](#)[Visual Studio](#)[WWF](#)[BizTalk Server](#)[Azure](#)[General](#)[Testing](#)[Online Certifications](#)[PHP](#)[My SQL](#)[LinQ](#)[Project Management](#)[Silverlight](#)[XML](#)[MS Office](#)[Windows OS](#)[DHTML](#)[Sharepoint](#)

[View All Articles](#)**Related Articles and Code:**

- [PROGRAM TO IMPLEMENT RECURSIVE DESCENT PARSER](#)
- [PROGRAM TO IMPLEMENT RECURSIVE DESCENT PARSER](#)
- [To parse a string using Recursive-Descent parser](#)
- [To parse a string using Recursive-Descent parser](#)
- [Example of recursive function](#)
- [Program to solve the Towers of Hanoi Problem \(using Recursive Algorithm\)](#)
- [Program to solve the Towers of Hanoi Problem \(using Recursive Algorithm\)](#)
- [Program to compute and displays the factorial of the given number \( using Recursive Algorithm \)](#)
- [Recursive program to generate fibonacci Series](#)
- [Recursive program for following operations fibonacci series](#)
- [Program to show an example of using a recursive method in a class](#)
- [Program of parser 1](#)
- [Program of parser 2](#)
- [Program of LL1 parser](#)
- [Program of parser 1](#)
- [To parse a string using First and Follow algorithm and LL-1 parser](#)
- [To parse a string using First and Follow algorithm and LL-1 parser](#)
- [Program of LL Parser](#)
- [Program of LL Parser](#)
- [Program that implements depth first search algorithm.](#)

**Other Interesting Articles in C++ Programming:**

- [Program that performs selection search](#)
- [Program to draw a Bezier Curve of nth degree](#)
- [Program to illustrate the functions returning pointers](#)
- [Program to perform quick sort](#)
- [Program to draw a 3D Cubic Bezier Curve](#)
- [Program to illustrate unary operator\(decrement operator\) overloading without return type](#)
- [Program that prints all the even numbers b/w 0 to 50 \( using while, do-while and for loop \)](#)
- [Program that defines template of vector class that provides modify and multiplication facility](#)
- [Program to fill different types of geometric shapes using Flood Fill Algorithm](#)
- [Write a program to Add Two Matrix](#)
- [Program that reads marks obtained by a student in a test of 100 marks and computes his grade](#)
- [Program to estimate the Differential value of a given function using Runge-Kutta Methods](#)
- [Program to illustrate the overloading of constructors in classes](#)
- [Program of address calculation sort](#)
- [Program of simple windows & buttons](#)
- [Program that reads an expression consisting of two non-negative integer and an operator. Determine if either integer or the result of the expres.....](#)
- [Program of displaying product inventory by converting one class to another class](#)
- [PROGRAM TO IMPLEMENT RECURSIVE DESCENT PARSER](#)
- [Program to convert an Infix Expression into a Postfix / Suffix Expression without parentheses](#)
- [Program to illustrate the implementation of 3D Rotation Transformation along x-axis](#)

**Please enter your Comment**

- Comment should be atleast 30 Characters.
- Please put code inside [Code] your code [/Code].

**Please login to post comment**

**No Comment Found, Be the First to post comment!**

**RSS Feeds:** [Articles](#) | [Forum](#) | [New Users](#) | [Activities](#) | [Interview FAQ](#) | [Poll](#) | [Hotlinks](#)

**Social Networking:** [Hall of Fame](#) | [Facebook](#) | [Twitter](#) | [LinkedIn](#)

**Terms:** [Terms of Use](#) | [Privacy Policy](#) | [Contact us](#)



Copyright © 2008-2012