Welcome - Guest! Login / Register Now |

Search: 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

To parse a string using Operator Precedence parsing

Posted By: Adalfreddo Fischer **Category:** C++ Programming **Views:** 13438

To parse a string using Operator Precedence parsing.

Code for To parse a string using Operator Precedence parsing in C++ Programming

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

int getOperatorPosition(char );

#define node struct tree1

int matrix[5][5]={
        {1,0,0,1,1},
        {1,1,0,1,1},
        {0,0,0,2,3},
        {1,1,3,1,1},
        {0,0,0,3,2}};
int tos=-1;
void matrix_value(void);
//node create_node(char,*node);void show_tree( node *);
int isOperator(char );

struct tree1
{
   char data;
   node  *lptr;
   node  *rptr;
}*first;


struct opr
{
    char op_name;
    node *t;
}oprate[50];


char cur_op[5]={'+','*','(',')','['};
char stack_op[5]={'+','*','(',')',']'};

void  main()
{
    char exp[10];

    int ssm=0,row=0,col=0;
    node *temp;
//    clrscr();

    printf("Enter Exp : ");
    scanf("%s",exp);

    matrix_value();
    while(exp[ssm] != '\0')
    {
        if(ssm==0)
        {
            tos++;
            oprate[tos].op_name = exp[tos];
        }
        else
        {
            if(isOperator(exp[ssm]) == -1)
            {
                oprate[tos].t = (node*) malloc (sizeof(node));
                oprate[tos].t->data = exp[ssm];
                oprate[tos].t->lptr = '\0';
```

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

```
                oprate[tos].t->rptr = '\0';
            }
            else
            {
                row = getOperatorPosition(oprate[tos].op_name);
                col = getOperatorPosition(exp[ssm]);
                if(matrix[row][col] == 0)
                {
                    tos++;
                    oprate[tos].op_name = exp[ssm];
                }
                elseif(matrix[row][col] == 1)
                {
                    temp = (node*) malloc (sizeof(node));
                    temp->data = oprate[tos].op_name;

                    temp->lptr = (oprate[tos-1].t);
                    temp->rptr = (oprate[tos].t);
                    tos--;

                    oprate[tos].t = temp;

                    ssm--;
                }
                elseif(matrix[row][col] == 2)
                {
                    //temp = (node*) malloc (sizeof(node));
                    temp = oprate[tos].t;
                    tos--;
                    oprate[tos].t = temp;
                }
                elseif(matrix[row][col] == 3)
                {
                                  printf("\nExpression is Invalid...\n");
   printf("%c  %c can not occur simultaneously\n",oprate[tos].op_name,exp[ssm]);
                    break;
                }
            }

        }

        ssm++;
    }
    printf("show tree \n\n\n");
    show_tree(oprate[tos].t);
    printf("Over");
    getch();
    getch();
}

int isOperator(char c)
{
    int i=0;
     for(i=0;i<5;i++)
     {
        if (c==cur_op[i] || c==stack_op[i])
            break;
     }

     if(i==5)
        return (-1);
     elsereturn i;

}

int getOperatorPosition(char c)
{
    int i;
    for(i=0;i<5;i++)
    {
        if (c==cur_op[i] || c==stack_op[i])
            break;
    }
    return i;

}

void show_tree(node *start)
{
    if(start->lptr != NULL)
        show_tree(start->lptr);

    if(start->rptr != NULL)
        show_tree(start->rptr);

    printf("%c \n",start->data);
}

void matrix_value(void)
{
     int i,j;
     printf("OPERATOR PRECEDENCE MATRIX\n");
     printf("===========================\n   ");
```

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

```
 for(i=0; i<5; i++)
 {
    printf("%c ",stack_op[i]);
 }
 printf("\n");

 for(i=0;i<5;i++)
 {
    printf("%c  ",cur_op[i]);
   for(j=0;j<5;j++)
   {
        if(matrix[i][j] == 0)
            printf("< ");
        elseif(matrix[i][j] == 1)
            printf("> ");
        elseif(matrix[i][j] == 2)
            printf("= ");
        elseif(matrix[i][j] == 3)
            printf("  ");
    }
    printf("\n");
 }

}
/************************************
              OUTPUT:
************************************/


Enter Exp : [a+b*c]
OPERATOR PRECEDENCE MATRIX
===========================
   + * ( ) ]
+  > < < > >
*  > > < > >
(  < < < =
)  > >   > >
[  < < <   =
show tree


a
b
c
*
+
Over
Enter Exp : [a+(b*c)+d]
OPERATOR PRECEDENCE MATRIX
===========================
   + * ( ) ]
+  > < < > >
*  > > < > >
(  < < < =
)  > >   > >
[  < < <   =
show tree


a
b
c
*
+
d
+
Over
Enter Exp : [)]
OPERATOR PRECEDENCE MATRIX
===========================
   + * ( ) ]
+  > < < > >
*  > > < > >
(  < < < =
)  > >   > >
[  < < <   =

Expression is Invalid...
[  ) can not occur simultaneously
show tree
.
Over

****************************************/
```

Share:

Previous Post:
To parse a string using First and Follow algorithm and LL-1 parser

Next Post:
To parse a string using Operator Precedence parsing

Didn't find what you were looking for? Find more on To parse a string using Operator Precedence parsing Or get search suggestion and latest updates.

Adalfreddo Fischer author of To parse a string using Operator Precedence parsing is from Frankfurt, Germany.

View All Articles

Related Articles and Code:

- To parse a string using Operator Precedence parsing
- To parse a string using Recursive-Descent parser
- To parse a string using Recursive-Descent parser
- 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 FOR LL(1) PARSING
- Program to show the implementation of Bottom-Up Parsing
- PROGRAM OF OPERATOR PRECEDENCE MATRIX
- Program to illustrate operator overloading from String object to basic string
- Program to copy one string s2 to another string s1 using strcopy function from string.h
- PROGRAM TO COPY ONE STRING INTO ANOTHER STRING WITH AND WITHOUT USING "MOVSB" INSTRUCTION USING MACRO
- Program to show the use of Comparison operator (==) for comparing Strings
- String Concatenation Operator (+)
- Program to illustrate operator overloading for strings
- Program to copy one string to another by coding and using string function
- Program to copy one string s2 to another string s1 without using strcopy function
- String operations like compare strings, concatenate strings, find length, reverse string, Find string and word in a file using while loop
- Program to compare strings, concatenate strings, copy string and display part of a string
- Program to print address of a variable using & operator
- Program to print address of a variable using & and * operators

Other Interesting Articles in C++ Programming:

- Program to add pairs of MxN matrices
- Program of doubly link list
- Program that sorts numbers using topological sort method
- Scanner of valide variable name in c
- Program to compute and print out the decimal notation for the reciprocal of each input integer
- Program that will allow a spy to encode and decode messages
- Identifer recognisition Integer Unsigned real number with optional integer part
- Program to illustrate the implementation of Scaling Transformation along a Fixed Point
- Program to covert a given Decimal Number into English Code
- Program to convert an Infix Expression into a Postfix Expression using Linked List as a Stack
- Files and Graphs
- Program to show the 3D Shearing Transformation along x-axis
- Program to evaluate series
- Program to illustrate the binary operator(*) overloading without creating a new object
- Program to illustrate unary operator(decrement operator) overloading without return type
- Program that prints the first 10 integers(using for loop)
- Program to create a circular linked list
- Program to represent or compress '2ABCDE' to string 'AABCDE''
- Program to illustrate the use of some builtin functions of header file &lt;math.h&gt;.
- Defines queue and priority queues. Lists basic operations and properties of queue. Also lists types of priority queue.

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