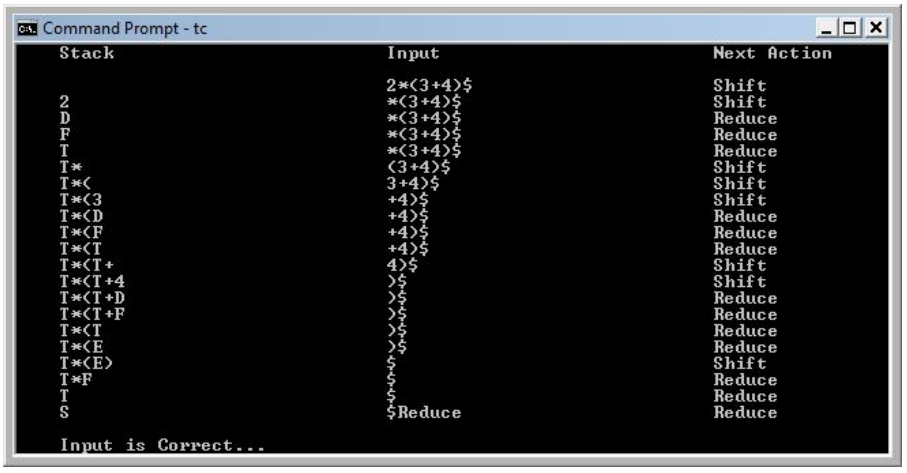
C++ Programming Articles

Program to show the implementation of Bottom-Up Parsing

Posted By: Easy Tutor **Category:** C++ Programming **Views:** 11739

A C++ Program to show the implementation of Bottom-Up Parsing.



Download Sourcecode for Program to show the implementation of Bottom-Up Parsing (Size: 1.33 KB)

Code for Program to show the implementation of Bottom-Up Parsing in C++ Programming

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

staticchar Stack[50][10]={NULL};
staticint top=-1;
staticint cit=0;

// Input Grammarstaticint productions[6]={5,1,7,7,2,10};

constchar Grammar[5][11][10]={
              {"S","E"},
              {"E","E+T","E-T","E*T","E/T","E%T","E^T","T"},
              {"T","T+F","T-F","T*F","T/F","T%F","T^F","F"},
              {"F","(E)","D"},
              {"D","0","1","2","3","4","5","6","7","8","9"}
            };

// Input Statementconstint input_length=8;
char Input[input_length][5]={"2","*","(","3","+","4",")","$"};


/*************************************************************************///--
{
   top++;

   strcpy(Stack[top],Token);
}

/*************************************************************************///--
{
   strset(Stack[top],NULL);

   top--;
}

/*************************************************************************///--
{
```

```
   for(int i=0;i<items;i++)
      Pop( );

   Push(Grammar[index][0]);
}

/*************************************************************************///--
{
   Push(Input[cit]);

   cit++;
}

/*************************************************************************///--
{
   int items;
   int index;

   char TopItems[100]={NULL};

   for(int i=0;i<=top;i++)
   {
      strset(TopItems,NULL);

      for(int j=i;j<=top;j++)
     strcat(TopItems,Stack[j]);

      for(j=0;j<productions[0];j++)
      {
     for(int k=1;k<=productions[(j+1)];k++)
     {
        if(strcmp(TopItems,Grammar[j][k])==0)
        {
       items=(top-i+1);
       index=j;

       goto NextCheck;
        }
     }
      }
   }

   return 0;

   NextCheck:

   char CitInput[20]={NULL};

   strcpy(CitInput,Stack[top]);
   strcat(CitInput,Input[cit]);

   for(i=0;i<productions[0];i++)
   {
      for(int j=1;j<=productions[(i+1)];j++)
      {
     if(strstr(Grammar[i][j],CitInput)!=NULL)
        return 0;
      }
   }

   Reduce(items,index);

   return 1;
}

/*************************************************************************//***
{
   clrscr( );

   int flag=0;

   cout<<"    /////*****++++++-----.....  Bottom-Up Parsing  .....-----++++++****

   gotoxy(5,3);
   cout<<"Stack";

   gotoxy(35,3);
   cout<<"Input";

   gotoxy(65,3);
   cout<<"Next Action";

   gotoxy(5,5);

   for(int i=0;i<=top;i++)
      cout<<Stack[i];

   gotoxy(35,5);

   for(int j=cit;j<input_length;j++)
      cout<<Input[j];

   gotoxy(65,5);
```

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

```
    cout<<"Shift";

    do
    {
       if(!CheckReduceCondition( ))
       {
      Shift( );

      gotoxy(65,(wherey( )+1));
      cout<<"Shift";
       }

       else
       {
      gotoxy(65,(wherey( )+1));
      cout<<"Reduce";
       }

       gotoxy(5,wherey( ));

       for(int i=0;i<=top;i++)
      cout<<Stack[i];

       gotoxy(35,wherey( ));

       for(int j=cit;j<input_length;j++)
      cout<<Input[j];

       if(top==0 && strcmp(Stack[top],Grammar[0][0])==0 &&
                          strcmp(Input[cit],"$")==0)
       {
      flag=1;

      break;
       }

       elseif(strcmp(Stack[top],"$")==0)
       {
      flag=0;
      break;
       }
    }
    while(1);

    if(flag)
       cout<<"\n\n    Input is Correct...";

    else
       cout<<"\n\n    Input is Incorrect...";

    getch( );
    return 0;
 }
[/Code]
```

Share:

Previous Post:
Program to illustrate self-referential structures "Linked List"

Next Post:
Program to convert an Infix Expression into a Postfix Expression using Linked List as a Stack

Didn't find what you were looking for? Find more on Program to show the implementation of Bottom-Up Parsing Or get search suggestion and latest updates.

Easy Tutor author of Program to show the implementation of Bottom-Up Parsing is from United States. Easy Tutor says

Hello Friends,

I am **Free Lance Tutor**, who helped student in completing their homework.

I have 4 Years of hands on experience on helping student in completing their homework. I also guide them in doing their final year projects.

I have share many programs on this website for everyone to use freely, **if you need further assistance, than please contact me on easytutor.2ya [at the rate] gmail [dot] com**

I have special discount scheme for providing tutor services. I am providing tutor service to students from various contries, currently most of my students are from United States, India, Australia, Pakistan, Germany, UK and Canada.

I am also here to expand my technical network to receive more opportunity in my career, make friends to help them in resolving their technical problem, learn and share my knowledge, If you like to be my friend, Please send me friend request.

Thanks,
Happy Programming :)

View All Articles

Related Articles and Code:

- **To parse a string using Operator Precedence parsing**
- **To parse a string using Operator Precedence parsing**
- **Program to show the Simple implementation of Hashing**
- **Program to show the implementation of Window-to-Viewport Coordinate Transformaton**
- **Program to show the implementation of Point Clipping Algorithm**
- **Program to show the implementation of Cohen-Sutherland Line Clipping Algorithm**
- **Program to show the implementation of Cohen-Sutherland MidPoint Subdivision Line Clipping Algorithm**
- **Program to show the implementation of Liang-Barsky Line Clipping Algorithm**
- **Program to show the implementation of Sutherland-Hodgeman Polygon Clipping Algorithm**
- **Program to show the implementation of Window-to-Viewport Transformaton**
- **Program to show the implementation of None-or-All String Clipping Strategy (Text Clipping Example)**
- **Program to show the implementation of None-or-All Character Clipping Strategy (Text Clipping Example)**
- **Program to show the implementation of Double Mouse Click**
- **Program to show the implementation of the Bubble Sort Algorithm**
- **Program to show the implementation of Binary Search Tree as Sets**
- **Program to show the implementation of Linked List as a Binary Search Tree**
- **PROGRAM FOR LL(1) PARSING**
- **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**

Other Interesting Articles in C++ Programming:

- **Program that implements overloaded "-" minus unary operator.**
- **Program to solve the Towers of Hanoi Problem (using Recursive Algorithm)**
- **Program to illustrate an example of Polymorphism (Pure Virtual functions).**
- **Write a function power() to raise a number m to a power n**
- **Program to show the implementation of Hashing with Insert, Delete and search functionality using double-ended Linked List**
- **Program to illustrate the implementation of Translation Transformation**
- **Program that reads in a matrix and checks if it has the parity property**
- **Program to computes the n_th term of the fibonacci series using Divide and Conquer Strategy**
- **Matrix Addition Program**
- **Program to draw a circle using MidPoint Circle Algorithm**
- **Program of Deapth First Search Traversal (DFS)**
- **Program to illustrate the relationship b/w union and structure**
- **Program to illustrate operator overloading for strings**
- **To parse a string using Recursive-Descent parser**
- **Program to illustrate the use of call-by-refrence method using pointers**
- **Program to determine whether the given function is a Cubic Spline or not**
- **Program to create a stack using dynamic memory allocation**
- **Program for function overloading**
- **PROGRAM FOR LL(1) PARSING**
- **Program to add two integers without using "+" operator**

Please enter your Comment

- Comment should be atleast 30 Characters.

- Please put code inside [Code] your code [/Code].

Please login to post comment

Easy Tutor from **United States** Comment on: Oct 10
It is Turbo C compiler.

Uchiha Budi from **Indonesia** Comment on: Oct 10
What compiler you used?

View All Comments