

C PROGRAM FOR RECURSIVE DESCENT PARSER

👤 🕒 March 3, 2013 📁 C, Compiler Design 👁 6,458 Views

C Program for implementation of Recursive Descent Parser for language given below

E-> TE'

E'-> +TE' | -TE' | null

T-> FT'

T'-> *FT' | /FT' | null

F-> id/ (E)/ num

Here 'id' can be a single letter. And 'num' can be a single digit number.

```

1  #include "stdio.h"
2  #include "conio.h"
3  char input[100];
4  char prod[100][100];
5  int pos=-1,l,st=-1;
6  char id,num;
7  void E();
8  void T();
9  void F();
10 void advance();
11 void Td();
12 void Ed();
13 void advance()
14 {
15 pos++;
16 if(pos<1)
17 {
18 if(input[pos]&gt;='0'&& input[pos]&lt;='9')
19 {
20 num=input[pos];
21 id='\0';
22 }
23 if((input[pos]&gt;='a' || input[pos]&gt;='A')&& (input[pos]&lt;='z' || input[pos]&lt;='Z'))
24 {id=input[pos];
25 num='\0';
26 }
27 }
28 }
29 void E()
30 {
31 strcpy(prod[++st],"E->TE'");
32 T();
33 Ed();
34 }
35 void Ed()
36 {
37 int p=1;
38 if(input[pos]=='+')
39 {
40 p=0;
41 strcpy(prod[++st],"E'->+TE'");
42 advance();
43 T();
44 Ed();
45 }
46 if(input[pos]=='-')
47 {
48 p=0;
49 strcpy(prod[++st],"E'->-TE'");
50 advance();
51 T();
52 Ed();
53 }
54 // Recursive Descent Parser
55 if(p==1)
56 {
57 strcpy(prod[++st],"E'->null");
58 }
59 }
60
61 void T()
62 {
63 strcpy(prod[++st],"T->FT'");
64 F();
65 Td();
66 }
67 void Td()
68 {

```

```

69 int p=1;
70 if(input[pos]=='*')
71 {
72     p=0;
73     strcpy(prod[++st],"T'-&gt;*FT'");
74     advance();
75     F();
76     Td();
77 }
78 if(input[pos]=='/')
79 {
80     p=0;
81     strcpy(prod[++st],"T'-&gt;/FT'");
82     advance();
83     F();
84     Td();
85 }
86 if(p==1)
87     strcpy(prod[++st],"T'-&gt;null");
88 void F()
89 {
90     if(input[pos]==id) {
91         strcpy(prod[++st],"F-&gt;id");
92         advance();
93     }
94     if(input[pos]=='(')
95     {
96         strcpy(prod[++st],"F-&gt;(E)");
97         advance();
98         E();
99         if(input[pos]==')') {
100             //strcpy(prod[++st],"F-&gt;(E)");
101             advance();
102         }
103     }
104     if(input[pos]==num)
105     {
106         strcpy(prod[++st],"F-&gt;num");
107         advance();
108     }
109 }
110 int main()
111 {

```

Stay updated to Programming and latest Events!

[Subscribe](#)

```

112 printf("Enter Input String : ");
113 scanf("%s",input);
114 l=strlen(input);
115 input[l]='\0';
116 advance();
117 E();
118 if(pos==l)
119 {
120     printf("String Accepted\n");
121     for(i=0;i<=st;i++)
122     {
123         printf("%s\n",prod[i]);
124     }
125 }
126 else
127 {
128     printf("String rejected\n");
129 }
130 getch();
131 return 0;
132 }

```

OUTPUT:

```

Enter Input String (a+b)
String Accepted
E->TE'
T->FT'
F->(E)
E->TE'
T->FT'
F->id
T'->null
E'->+TE'
T->FT'
F->id
T'->null
E'->null
T'->null
E'->null

```

If you like this then please share this page. You can comment you queries here. Or you can also ask for a program. We are happy to help you.

tags: C program for Recursive Decsent parser, compiler design programs, parsing techniques

