

```
1  #include<stdio.h>
2  #include<conio.h>
3  int isterm(char x)
4  {
5      if((x=='a')||(x=='+')||(x=='*')||(x=='(')||(x==')')||(x=='$'))
6          return 1;
7      else
8          return 0;
9  }
10 void main()
11 {
12     char table[5][6][4]={{ "TA", "", "", "TA", "", ""},
13         {"", "+TA", "", "", "e", "e"},
14         {"FB", "", "", "FB", "", ""},
15         {"", "e", "*FB", "", "e", "e"},
16         {"a", "", "", "(E)", "", ""}};
17     char stack[20], input[20];
18     int len, l, i, j, k, nt, t, flag;
19     clrscr();
20     printf("Enter the input string to be parsed : ");
```

main.c

```
22 l=strlen(input);
23 input[l]='$';
24 input[l+1]='\0';
25 stack[0]='$';
26 stack[1]='E';
27 stack[2]='\0';
28 i=1;j=0;
29 printf("\n\n");
30 printf("-----\n");
31 printf("STACK\tINPUT\tOUTPUT\n");
32 printf("-----\n");
33 printf("%s\t%s\n",stack,input);
34 while(!(stack[i]=='$'&&input[0]=='$'))
35 {
36     if((stack[i]=='$')&&(input[0]!='$'))
37     {
38         printf("Not a sentence\n");
39         goto a;
40     }
41     else if(isterm(stack[i])&&(stack[i]!=input[0]))
```

input


```
49     l--;  
50     l=strlen(input);  
51     for(j=0;j<l-1;j++)  
52     {  
53         input[j]=input[j+1];  
54     }  
55     input[l-1]='\0';  
56     flag=0;  
57 }  
58 else  
59 {  
60     switch(stack[i])  
61     {  
62         case 'E': nt=0;break;  
63         case 'A': nt=1;break;  
64         case 'T': nt=2;break;  
65         case 'B': nt=3;break;  
66         case 'F': nt=4;break;  
67     }  
68     switch(input[0])  
69     {
```

main.c

```
70     case 'a': t=0; break;
71     case '+': t=1; break;
72     case '*': t=2; break;
73     case '(': t=3; break;
74     case ')': t=4; break;
75     case '$': t=5; break;
76 }
77 stack[i]='\0';
78 i--;
79 l=strlen(table[nt][t]);
80 if(l==0)
81 {
82     printf("Not a sentence\n");
83     goto a;
84 }
85 else if(strcmp(table[nt][t],"e")==0)
86 {
87 }
88 else
89 {
90     flag=1;
```




Run



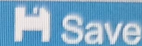
Debug



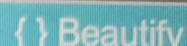
Stop



Share



Save



{ } Beautify



main.c

```
88         else
89         {
90             flag=1;
91             for(k=l-1;k>=0;k--)
92             {
93                 stack[++i]=table[nt][t][k];
94             }
95         }
96     }
97     if(flag==0)
98     printf("%s\t%s\n",stack,input);
99     else
100     printf("%s\t%s\t%s\n",stack,input,table[nt][t]);
101 }
102 printf("Sentence\n");
103 a:
104 getch();
105 }
106
107
```

C:\Users\srija\OneDrive\Desktop\cd5.exe

Implementation of RECURSIVE DESCENT PARSER

Until

1 Enter the Expression:

2 a

3

4 The string is rejected.

5

6

7

8