```
1 /* ///*
           * ASSIGNMENT: LL(1) PARSER
         * AUTHORS: BIJAY KHATRI & DEEPAK VERMA
         * ROLL: 12/CS/45 AND 12/CS/46
        *////*/
    6 #include <string.h>
         #include <stdio.h>
   8
         #include <stdlib.h>
   9 int main()
 10 {
               \textbf{int} \ i=0, j=0, k=0, m=0, n=0, o=0, o1=0, var=0, l=0, f=0, c=0, f1=0;
11
               char str[30],str1[40]="E",temp[20],temp1[20],temp2[20],tt[20],t3[20];
 12
 13
               strcpy(temp1,"\0");
 14
               strcpy(temp2,"\setminus0");
 15
               char t[10];
 16
               char array[6][5][10] = {
                              "NT", "<id>","+","*",",
"E", "Te","Error","Error","Error",
"e", "Error","+Te","Error","\0",
17
 18
19
                              "T", "Vt", "Error", "Error", "Error",
"t", "Error", "\0", "*Vt", "\0",
"V", "<id>>", "Error", "Error",
20
 21
22
23
               printf("\n\tLL(1) PARSER TABLE \n");
 24
 25
               for(i=0;i<6;i++)
26
 27
                    for(j=0;j<5;j++)
28
 29
                        printf("%10s",array[i][j]);
30
31
                   printf("\n");
32
 33
               printf("\n\tENTER THE STRING:");
34
               gets(str);
35
               if(str[strlen(str)-1]!=';')
36
37
                 printf("END OF STRING MARKER SHOULD BE ';'");
38
                exit(1);
39
40
               printf("\n\tCHECKING VALIDATION OF THE STRING ");
41
               printf("\n\t%s",str1);
42
43
 44
          while(i<strlen(str))
45
              {
 46
               again:
                     if(str[i] == ' ' && i<strlen(str))</pre>
47
 48
49
                            printf("\n\tSPACES IS NOT ALLOWED IN SOURSE STRING ");
50
                            exit(1);
 51
52
                      temp[k]=str[i];
53
                     temp[k+1]='\backslash 0';
54
                     f1=0;
55
                again1:
56
                     if(i>=strlen(str))
57
58
                            exit(1);
59
 60
                      for(int l=1;l<=4;l++)
61
                        if(strcmp(temp,array[0][l])==0)
 62
63
                        {
64
                             f1=1;
65
                             m=0,o=0,var=0,o1=0;
 66
                             strcpy(temp1,"\0");
                             strcpy(temp2,"\setminus0");
67
                             int len=strlen(str1);
68
                             while(m<strlen(str1) && m<strlen(str))</pre>
69
70
 71
                                     if(str1[m]==str[m])
72
 73
                                           var=m+1;
 74
                                           temp2[o1]=str1[m];
 75
                                           m++;
76
                                           01++;
77
                                    }
```

```
78
                  else
 79
 80
                    if((m+1)<strlen(str1))</pre>
 81
 82
 83
                      temp1[o]=str1[m];
 84
 85
                    }
 86
                    else
 87
                      m++;
 88
                 }
 89
 90
 91
              temp2[o1] = '\0';
              temp1[o] = ' \setminus 0';
 92
              t[0] = str1[var];
 93
 94
              t[1] = '\setminus 0';
 95
              for(n=1;n<=5;n++)
 96
 97
                 if(strcmp(array[n][0],t)==0)
 98
                   break;
 99
100
              strcpy(str1,temp2);
101
              strcat(str1,array[n][l]);
102
              strcat(str1,temp1);
103
              printf("\n\t%s",str1);
104
              \textbf{if}(strcmp(\textbf{array}[n][l],"\backslash 0") == 0)
105
106
                 if(i==(strlen(str)-1))
107
108
                {
109
                    int len=strlen(str1);
110
                    str1[len-1]='\0';
                    printf("\n\t%s",str1);
111
112
                    printf("\n\n\tENTERED STRING IS VALID");
                    exit(1);
113
114
                 strcpy(temp1,"\setminus0");
115
116
                 strcpy(temp2,"\setminus0");
117
                 strcpy(t,"\setminus 0");
118
                  goto again1;
119
120
              if(strcmp(array[n][l],"Error")==0)
121
               printf("\n\tERROR IN YOUR SOURCE STRING");
122
123
                 exit(1);
124
125
              strcpy(tt,"\setminus 0");
126
              strcpy(tt,array[n][l]);
127
              strcpy(t3,"\setminus0");
128
              f=0;
129
              for(c=0;c<strlen(tt);c++)</pre>
130
131
                 t3[c]=tt[c];
132
                 t3[c+1]='\backslash 0';
133
                 if(strcmp(t3,temp)==0)
134
135
                    f=0;
136
                    break;
137
138
                 else
139
                    f=1;
140
               }
141
142
               if(f==0)
143
144
                strcpy(temp,"\setminus 0");
                strcpy(temp1,"\0");
145
                strcpy(temp2,"\setminus0");
146
147
                strcpy(t,"\setminus 0");
148
                i++;
149
                k=0;
150
                goto again;
151
152
               else
153
154
                strcpy(temp1,"\setminus0");
```

```
strcpy(temp2,"\setminus0");
155
156
          strcpy(t,"\setminus 0");
157
           goto again1;
158
159
       }
160
       }
161
       i++;
162
       k++;
163
164
     if(f1==0)
       printf("\nENTERED STRING IS INVALID");
165
166
        printf("\n\text{ENTERED STRING IS VALID"});
167
168 }
169
170
171
172
173
174
176
177 * LL(1) PARSER TABLE
178
      179
180
181
        XXX +Te XXX
      e
          Vt XXX XXX XXX
182
      \mathbf{T}
183
      t XXX
                 *Vt
         <id> XXX XXX XXX
184
185
186
     ENTER THE STRING: <id>+<id>*<id>
187
     CHECKING VALIDATION OF THE STRING
188
189
190
     Te
191
     Vte
192
     <id>te
193
     <id>e
194
     <id>+Te
195
     <id>+Vte
196
     <id>+<id>te
197
     <id>+<id>*Vte
198
     <id>+<id>*<id>te
199
     <id>+<id>*<id>e
200
     <id>+<id>*<id>
201
202
     ENTERED STRING IS VALID
203
204
206
```