

```

1 digit [0-9]+
2 double [0-9]+[.][0-9]+
3 string [A-Za-z]+
4
5 datatype "int"|"float"|"char"
6 variable [A-Za-z]+[0-9]*[A-Za-z]*
7 value {digit}|{string}|{double}|{variable}
8 array {variable}"["{digit}"]"
9 frtype "void" | {datatype}
10 keyword
"include"|"while"|"loop"|"if"|"elif"|"else"|"switch"|"return" | {frtype}|"func"|"end_func"|"end_if"|"end_main"
11
12 operator "+"|"-"|"*"|"/"|"%"|"^"|"&"|"|"
13 relational "<"|"="|"&&"|"||"|">"|"<="|">="|"!="
14 allof {relational}|{operator}
15
16 assign {variable}[=]{value}|{variable}[=]{value}{operator}{value}
17 expr {value}{allof}{value}
18 condition {expr}({relational}{expr})*
19 identifier {variable}|{array}
20 header {string}".h;"
21 coma [,]
22 space [ ]
23 parameter {datatype}{space}{variable}({coma}{datatype}{space}{variable})*
24
25 %{
26 #include<string.h>
27 #include<stdio.h>
28 int cm=0;
29 int varc=0,functc=0,opc=0,arrc=0,keyc=0,headc=0;
30 int funs=-1;
31 int iff=-1,eliff=-1,elsef=-1,lop=-1,whl=-1,ms=0;
32 %}
33
34 %%
35 "/*".* {printf("Single line comment\n");}
36 "/*".* { if(cm==0) { cm=1; printf("Multi-line comment\n"); } }
37 "*/" {if(cm){ cm=0; printf("multi-line comment ended\n");}
38     else{ printf("%s",yytext); }}
39
40 "#include"[ ]{keyword}".h;" { if(cm==0) { printf("Error : header file name can't be keyword\n"); } }
41 "#include"[ ]{string}".h;" { if(cm==0) { printf("Header file included\n"); headc++; keyc++; } }
42 "#include".* {if(cm==0) printf("Error : Header file is not included\n");}
43
44 "func int main()" { if(cm==0){
45     if(ms==0){
46         ms=1; keyc+=3;
47         printf("Main function starts\n");
48     }
49     else{ printf("Error: a program can have only one main function\n"); } } }
50 "end_main;" { if(cm==0){
51     if(ms==1){ printf("Main function ends\n"); keyc++;}
52     else{printf("Main function does not exist\n");} } }
53 "end_func;" { if(cm==0){
54     if(funs==1){ printf("Error : No function starts\n"); }
55     else if(funs==1) {
56         funs=-1; keyc++; printf("function ends\n");
57     } } }
58 "func"[ ]{frtype}[ ]{keyword}.* { if(cm==0) {
59 printf("Error : function name can't be keyword\n"); } }
60 "func"[ ]{frtype}[ ]{variable}"("{parameter}?" )" { if(cm==0){
61     if(funs==1){ funs=1;functc++;keyc+=2; printf("Function Declared\n");}
62     else{ printf("Error : A function can't be declared inside another function\n");} } }
63 {variable}"("{value}[","{value}]*)?" )" { if(cm==0){
64     printf("function called\n"); } }
65

```

```

66  "elif("{condition}")" { if(cm==0){
67      if(iff==1){ keyc++;
68          printf("elif clause found\n");}
69      else { printf("Error : No if clause found"); } } }
70  "if("{condition}")" { if(cm==0){
71      if(iff==1){
72          iff=1; keyc++;
73          printf("if clause found\n"); }
74      else { printf("Error : nested if is not possible\n"); } } }
75  "else" { if(cm==0){ if(iff==1){ keyc++; printf("else clause found\n"); }
76      else { printf("Error : No if clause found"); } } }
77  "end_if;" { if(cm==0){ if(iff==1){
78      iff=-1; keyc++;
79      printf("End of if clause\n"); }
80      else { printf("Error : No if clause found"); } } }
81  "end_while;" {if(cm==0){
82      if(whl==1){ keyc++; whl=-1; printf("while loop ends\n");}
83      else { printf("No while loop declared"); } } }
84  "while("{condition}")" {if(cm==0){
85      if(whl==1){
86          whl=1; keyc++; printf("while loop starts\n");}
87      else { printf("Error: Nested loop is not possible\n"); } }}
88  "end_loop;" {if(cm==0){
89      if(lop==1){ keyc++;lop=-1; printf("Loop ends\n"); }
90      else { printf("Error : No loop declared"); } } }
91  "loop"[ ]{digit}".."{digit}[ ]"by"[ ]{digit} { if(cm==0){
92      if(lop==1){ lop=1;keyc++; printf("Loop found\n");}
93      else { printf("Error: Nested loop is not possible\n"); } } }
94
95  {datatype}" "{variable}([,]{variable})*";" { if(cm==0){ printf("variable declared\n");keyc++; int i;
96  for( i=0;i<yy leng;i++){
97  if(yytext[i]==',' || yytext[i]==';') varc++; } }}
98
99  {datatype}" "{array}([,]{array})*";" { if(cm==0){ printf("array declared\n");keyc++; int i;
100 for( i=0;i<yy leng;i++){
101 if(yytext[i]==',' || yytext[i]==';') arrc++; } } }
102
103 {assign}[;] {if(cm==0) {printf("Value assigned\n"); } }
104 "return"[ ]{value}[;] {if(cm==0) {printf("Value returned\n"); } }
105
106 %%
107
108 int yywrap()
109 {
110     return 1;
111 }
112
113 int main()
114 {
115     yyin = freopen("in.txt","r",stdin);
116     yyout = freopen("out.txt","w",stdout);
117     yylex();
118     printf("Count : \n\tHeader : %d\n\tFunction : %d\n\tKeyword : %d\n\tVariable : %d\n\tArray :
%d\n",headc,functc,keyc,varc,arrc);
119     return 0;
120 }

```