```c
#include "token.h"
#include <stdlib.h>
#include "regstack.h"

#define MAXREGSIZE 8
#define MAXARGS 4
#define MAXLBLS 10

TOKEN* stack[MAXREGSIZE];
TOKEN* arg_stack[MAXARGS];
TOKEN* lbl_stack[MAXLBLS];

int top = -1;
int top_args = -1;
int top_lbls = -1;

int isempty() {

   if(top == -1)
      return 1;
   else
      return 0;
}

int isempty_args() {

   if(top_args == -1)
      return 1;
   else
      return 0;
}

int isempty_lbls() {

   if(top_lbls == -1)
      return 1;
   else
      return 0;
}

int isfull() {

   if(top == MAXREGSIZE)
      return 1;
   else
      return 0;
}

int isfull_args() {

   if(top_args == MAXARGS)
      return 1;
   else
      return 0;
}

int isfull_lbls() {

   if(top_lbls == MAXLBLS)
      return 1;
```

```
61    else
62       return 0;
63 }
64 
65 TOKEN* pop() {
66    TOKEN* data;
67 
68    if(!isempty()) {
69       data = stack[top];
70       top = top - 1;
71       return data;
72    } else {
73       return NULL;
74    }
75 }
76 
77 TOKEN* peep(){
78     TOKEN* data;
79     if(!isempty()) {
80       data = stack[top];
81       return data;
82    } else {
83       return NULL;
84    }
85 }
86 
87 TOKEN* peep_lbl(){
88     TOKEN* data;
89     if(!isempty_lbls()) {
90       data = lbl_stack[top_lbls];
91       return data;
92    } else {
93       return NULL;
94    }
95 }
96 
97 int push(TOKEN* data) {
98 
99    if(!isfull()) {
100      top = top + 1;
101      stack[top] = data;
102      return 0;
103   } else {
104      return -1;
105   }
106 }
107 
108 int push_arg(TOKEN* data) {
109 
110 if(!isfull_args()) {
111   top_args = top_args + 1;
112   arg_stack[top_args] = data;
113   return 0;
114 } else {
115   return -1;
116 }
117 }
118 
119 TOKEN* pop_arg() {
120   TOKEN* data;
```

```
121
122    if(!isempty_args()) {
123       data = arg_stack[top_args];
124       top_args = top_args - 1;
125       return data;
126    } else {
127       return NULL;
128    }
129 }
130
131 int push_lbl(TOKEN* data) {
132
133 if(!isfull_lbls()) {
134    top_lbls = top_lbls + 1;
135    lbl_stack[top_lbls] = data;
136    return 0;
137 } else {
138    return -1;
139 }
140 }
141
142 TOKEN* pop_lbl() {
143    TOKEN* data;
144
145    if(!isempty_lbls()) {
146       data = lbl_stack[top_lbls];
147       top_lbls = top_lbls - 1;
148       return data;
149    } else {
150       return NULL;
151    }
152 }
```