

```

1  #include <stdio.h>
2  #include <stdlib.h>
3
4  #define MAX_SIZE 100
5
6  struct Stack {
7      int top;
8      char items[MAX_SIZE];
9  };
10
11 void initialize(struct Stack *stack);
12 int isEmpty(const struct Stack *stack);
13 int isFull(const struct Stack *stack);
14 void push(struct Stack *stack, char item);
15 char pop(struct Stack *stack);
16 int isMatching(char opening, char closing);
17 void checkParenthesesErrors(const char *filename);
18
19 int main() {
20     char filename[100];
21
22     printf("Enter the filename: ");
23     scanf("%s", filename);
24
25     checkParenthesesErrors(filename);
26
27     return 0;
28 }
29
30 // To initialize the stack
31 void initialize(struct Stack *stack) {
32     stack->top = -1;
33 }
34
35 // To check if the stack is empty
36 int isEmpty(const struct Stack *stack) {
37     return stack->top == -1;
38 }
39
40 // To check if the stack is full
41 int isFull(const struct Stack *stack) {
42     return stack->top == MAX_SIZE - 1;
43 }
44
45 // To push an element onto the stack
46 void push(struct Stack *stack, char item) {
47     if (isFull(stack)) {
48         printf("Stack is full\n");
49         exit(EXIT_FAILURE);
50     }
51     stack->items[++stack->top] = item;
52 }
53
54 // To pop an element from the stack
55 char pop(struct Stack *stack) {
56     if (isEmpty(stack)) {
57         printf("Stack is empty\n");
58         exit(EXIT_FAILURE);
59     }
60     return stack->items[stack->top--];
61 }
62
63 // To check if two parentheses form a matching pair
64 int isMatching(char opening, char closing) {
65     return (opening == '(' && closing == ')') ||
66         (opening == '{' && closing == '}') ||

```

```

67         (opening == '[' && closing == ']');
68     }
69
70     // To check for parentheses errors in the code
71     void checkParenthesesErrors(const char *filename) {
72         FILE *file = fopen(filename, "r");
73         if (file == NULL) {
74             perror("File not found");
75             exit(EXIT_FAILURE);
76         }
77
78         struct Stack stack;
79         initialize(&stack);
80
81         char ch;
82         int line = 1;
83
84         // To read file
85         while ((ch = fgetc(file)) != EOF) {
86             if (ch == '\n') {
87                 line++;
88             } else if (ch == '(' || ch == '{' || ch == '[') {
89                 push(&stack, ch);
90             } else if (ch == ')' || ch == '}' || ch == ']') {
91                 if (isEmpty(&stack) || !isMatching(pop(&stack), ch)) {
92                     if (isEmpty(&stack)) {
93                         printf("Extra %c at line %d\n", ch, line);
94                     } else {
95                         printf("Mismatched %c at line %d\n", ch, line);
96                     }
97                 }
98             }
99         }
100
101         // To check for any remaining opening parentheses
102         while (!isEmpty(&stack)) {
103             printf("Missing %c at line %d\n", pop(&stack), line);
104         }
105
106         fclose(file);
107     }

```