1. Variable Declarations

Input: Simple variable declaration using standard data types.

int x = 10;

float y = 3.14;

char c = 'A';

```
Enter your C-like code (type 'END' on a new line to finish input):
int x = 10;
float y = 3.14;
char c = 'A';
END
Input:
int x = 10;
float y = 3.14;
char c = 'A';

Tokenized Output:
Token: int, Type: Keyword
Token: x, Type: Identifier
Token: =, Type: Operator
Token: 10, Type: Number
Token: ;, Type: Symbol
Token: float, Type: Keyword
Token: y, Type: Identifier
Token: =, Type: Operator
Token: 3.14, Type: Number
Token: ;, Type: Symbol
Token: char, Type: Keyword
Token: c, Type: Identifier
Token: =, Type: Operator
Token: ', Type: Unknown
Token: A, Type: Identifier
Token: ', Type: Unknown
Token: ;, Type: Symbol
Syntax Analysis:
Syntax analysis completed successfully.

PS C:\Users\uditi\AppData\Local\Microsoft\Windows\INetCache\IE\Y637SSUO\output>
```

Compiled successfully!

2. Simple Assignment Statements

Input: Variable assignments after declaration.

int x = 10;

x = x + 5;

```
PS C:\Users\uditi\AppData\Local\Microsoft\Windows\INetCache\IE\Y637SSUO\output> & .\'syntax_analysis.exe'
Enter your C-like code (type 'END' on a new line to finish input):
int x = 10;
x = x + 5;
END
Input:
int x = 10;
x = x + 5;

Tokenized Output:
Token: int, Type: Keyword
Token: x, Type: Identifier
Token: =, Type: Operator
Token: 10, Type: Number
Token: ;, Type: Symbol
Token: x, Type: Identifier
Token: =, Type: Operator
Token: x, Type: Identifier
Token: +, Type: Operator
Token: 5, Type: Number
Token: ;, Type: Symbol
Syntax Analysis:
Syntax analysis completed successfully.

PS C:\Users\uditi\AppData\Local\Microsoft\Windows\INetCache\IE\Y637SSUO\output>
```

Compiled successfully!

3. Conditional Statements

Input: Handling if conditions.

int x = 10;

if (x > 5) x = x + 1;

```
Enter your C-like code (type 'END' on a new line to finish input):
int x = 10;
if (x > 5) x = x + 1;
END
Input:
int x = 10;
if (x > 5) x = x + 1;

Tokenized Output:
Token: int, Type: Keyword
Token: x, Type: Identifier
Token: =, Type: Operator
Token: 10, Type: Number
Token: ;, Type: Symbol
Token: if, Type: Keyword
Token: (, Type: Symbol
Token: x, Type: Identifier
Token: >, Type: Operator
Token: 5, Type: Number
Token: ), Type: Symbol
Token: x, Type: Identifier
Token: =, Type: Operator
Token: x, Type: Identifier
Token: +, Type: Operator
Token: 1, Type: Number
Token: ;, Type: Symbol
Syntax Analysis:
Syntax analysis completed successfully.

PS C:\Users\uditi\AppData\Local\Microsoft\Windows\INetCache\IE\Y637SSUO\output>
```

Compiled successfully!

4. Multiple Statements

Input: Multiple statements inside a conditional.

int a = 20;

```
if (a < 30) {
    a = a + 1;
}
```

```
END
Input:
int a = 20;
if (a < 30) {
    a = a + 1;
}

Tokenized Output:
Token: int, Type: Keyword
Token: a, Type: Identifier
Token: =, Type: Operator
Token: 20, Type: Number
Token: ;, Type: Symbol
Token: if, Type: Keyword
Token: (, Type: Symbol
Token: a, Type: Identifier
Token: <, Type: Operator
Token: 30, Type: Number
Token: ), Type: Symbol
Token: {, Type: Symbol
Token: a, Type: Identifier
Token: =, Type: Operator
Token: a, Type: Identifier
Token: +, Type: Operator
Token: 1, Type: Number
Token: ;, Type: Symbol
Token: }, Type: Symbol
Syntax Analysis:
Syntax analysis completed successfully.

PS C:\Users\uditi\AppData\Local\Microsoft\Windows\INetCache\IE\Y637SSUO\output>
```

5. Looping Statements (Simple Form)

Input: Handling while loops.

```
int i = 0;
while (i < 10) {
    i = i + 1;
}
```

```
END
Input:
int i = 0;
while (i < 10) {
    i = i + 1;
}

Tokenized Output:
Token: int, Type: Keyword
Token: i, Type: Identifier
Token: =, Type: Operator
Token: 0, Type: Number
Token: ;, Type: Symbol
Token: while, Type: Keyword
Token: (, Type: Symbol
Token: i, Type: Identifier
Token: <, Type: Operator
Token: 10, Type: Number
Token: ), Type: Symbol
Token: {, Type: Symbol
Token: i, Type: Identifier
Token: =, Type: Operator
Token: i, Type: Identifier
Token: +, Type: Operator
Token: 1, Type: Number
Token: ;, Type: Symbol
Token: }, Type: Symbol
Syntax Analysis:
Syntax analysis completed successfully.

PS C:\Users\uditi\AppData\Local\Microsoft\Windows\INetCache\IE\Y637SSUO\output>
```

Compiled successfully!

6. Invalid Declarations (Syntax Error)

Input: Missing identifier in a declaration.

int = 10;

```
PS C:\Users\uditi\AppData\Local\Microsoft\Windows\INetCache\IE\Y637SSUO\output> & .\'syntax_analysis.exe'
Enter your C-like code (type 'END' on a new line to finish input):
int = 10;
END
Input:
int = 10;

Tokenized Output:
Token: int, Type: Keyword
Token: =, Type: Operator
Token: 10, Type: Number
Token: ;, Type: Symbol
Syntax Analysis:
Syntax Error: Expected identifier before '='

PS C:\Users\uditi\AppData\Local\Microsoft\Windows\INetCache\IE\Y637SSUO\output>
```

7. Invalid Assignment (Syntax Error)

Input: Missing = in an assignment.

int x;

x 10;

```
Enter your C-like code (type 'END' on a new line to finish input):
int x;
x 10;
END
Input:
int x;
x 10;

Tokenized Output:
Token: int, Type: Keyword
Token: x, Type: Identifier
Token: ;, Type: Symbol
Token: x, Type: Identifier
Token: 10, Type: Number
Token: ;, Type: Symbol
Syntax Analysis:
Syntax Error: Expected '=' after identifier 'x'

PS C:\Users\uditi\AppData\Local\Microsoft\Windows\INetCache\IE\Y637SSUO\output>
```

8. Missing Semicolon (Syntax Error)

Input: Missing semicolon at the end of an assignment.

int x = 10

```
PS C:\Users\uditi\AppData\Local\Microsoft\Windows\INetCache\IE\Y637SSUO\output> & .\'syntax_analysis.exe'
Enter your C-like code (type 'END' on a new line to finish input):
int x = 10
END
Input:
int x = 10

Tokenized Output:
Token: int, Type: Keyword
Token: x, Type: Identifier
Token: =, Type: Operator
Token: 10, Type: Number
Syntax Analysis:
Syntax Error: Expected ';' at the end of the statement

PS C:\Users\uditi\AppData\Local\Microsoft\Windows\INetCache\IE\Y637SSUO\output>
```

9. Missing Parenthesis (Syntax Error in Condition)

Input: Incorrect parenthesis usage in a conditional.

if x > 5) x = x + 1;

```
Enter your C-like code (type 'END' on a new line to finish input):
if x > 5) x = x + 1;
END
Input:
if x > 5) x = x + 1;

Tokenized Output:
Token: if, Type: Keyword
Token: x, Type: Identifier
Token: >, Type: Operator
Token: 5, Type: Number
Token: ), Type: Symbol
Token: x, Type: Identifier
Token: =, Type: Operator
Token: x, Type: Identifier
Token: +, Type: Operator
Token: 1, Type: Number
Token: ;, Type: Symbol
Syntax Analysis:
Syntax Error: Unmatched closing parenthesis ')'

PS C:\Users\uditi\AppData\Local\Microsoft\Windows\INetCache\IE\Y637SSUO\output>
```

Compiled successfully!

10. Unclosed Block (Syntax Error in Loops/Conditionals)

Input: Unclosed braces for a block statement.

while (i < 10) {

  i = i + 1;

```
Enter your C-like code (type 'END' on a new line to finish input):
while (i < 10) {
    i = i + 1;
END
Input:
while (i < 10) {
    i = i + 1;

Tokenized Output:
Token: while, Type: Keyword
Token: (, Type: Symbol
Token: i, Type: Identifier
Token: <, Type: Operator
Token: 10, Type: Number
Token: ), Type: Symbol
Token: {, Type: Symbol
Token: i, Type: Identifier
Token: =, Type: Operator
Token: i, Type: Identifier
Token: +, Type: Operator
Token: 1, Type: Number
Token: ;, Type: Symbol
Syntax Analysis:
Syntax Error: Unmatched opening brace '{'

PS C:\Users\uditi\AppData\Local\Microsoft\Windows\INetCache\IE\Y637SSUO\output>
```

11. Simple Variable Declaration

int a;

float b;

```
PS C:\Users\uditi> cd 'c:\Users\uditi\AppData\Local\Microsoft\Windows\INetCache\IE\Y637SSUO\output'
PS C:\Users\uditi\AppData\Local\Microsoft\Windows\INetCache\IE\Y637SSUO\output> & .\'syntax_analysis.exe'
Enter your C-like code (type 'END' on a new line to finish input):
int a;
float b;
END
Input:
int a;
float b;

Tokenized Output:
Token: int, Type: Keyword
Token: a, Type: Identifier
Token: ;, Type: Symbol
Token: float, Type: Keyword
Token: b, Type: Identifier
Token: ;, Type: Symbol
Syntax Analysis:
Syntax analysis completed successfully.

PS C:\Users\uditi\AppData\Local\Microsoft\Windows\INetCache\IE\Y637SSUO\output>
```

12. Unmatched Parentheses

if (x > 0) {

printf("Positive");

```
PS C:\Users\uditi\AppData\Local\Microsoft\Windows\INetCache\IE\Y637SSUO\output> & .\'syntax_analysis.exe'
Enter your C-like code (type 'END' on a new line to finish input):
if (x > 0) {
    printf("Positive");
END
Input:
if (x > 0) {
    printf("Positive");

Tokenized Output:
Token: if, Type: Keyword
Token: (, Type: Symbol
Token: x, Type: Identifier
Token: >, Type: Operator
Token: 0, Type: Number
Token: ), Type: Symbol
Token: {, Type: Symbol
Token: printf, Type: Identifier
Token: (, Type: Symbol
Token: ", Type: Unknown
Token: Positive, Type: Identifier
Token: ", Type: Unknown
Token: ), Type: Symbol
Token: ;, Type: Symbol
Syntax Analysis:
Syntax Error: Unmatched opening brace '{'

PS C:\Users\uditi\AppData\Local\Microsoft\Windows\INetCache\IE\Y637SSUO\output>
```

13. Assignment Without Identifier

int = 10;

```
PS C:\Users\uditi\AppData\Local\Microsoft\Windows\INetCache\IE\Y637SSUO\output> & .\'syn.exe'
Enter your C-like code (type 'END' on a new line to finish input):
int = 10;
END
Input:
int = 10;

Tokenized Output:
Token: int, Type: Keyword
Token: =, Type: Operator
Token: 10, Type: Number
Token: ;, Type: Symbol
Syntax Analysis:
Syntax Error: Operator '=' found without preceding identifier.
```

14 Complex Expression

int sum = a + b * c / e - d;

```
int sum = a + b * c / e - d;
END
Input:
int sum = a + b * c / e - d;

Tokenized Output:
Token: int, Type: Keyword
Token: sum, Type: Identifier
Token: =, Type: Operator
Token: a, Type: Identifier
Token: +, Type: Operator
Token: b, Type: Identifier
Token: *, Type: Operator
Token: c, Type: Identifier
Token: /, Type: Operator
Token: e, Type: Identifier
Token: -, Type: Operator
Token: d, Type: Identifier
Token: ;, Type: Symbol
Syntax Analysis:
Syntax analysis completed successfully.

PS C:\Users\uditi\AppData\Local\Microsoft\Windows\INetCache\IE\Y637SSUO\output>
```

15. Complex Expression

int a = b + c – d * e;

```
Enter your C-like code (type 'END' on a new line to finish input):
int a= b+ c- d* e;
END
Input:
int a= b+ c- d* e;

Tokenized Output:
Token: int, Type: Keyword
Token: a, Type: Identifier
Token: =, Type: Operator
Token: b, Type: Identifier
Token: +, Type: Operator
Token: c, Type: Identifier
Token: -, Type: Operator
Token: d, Type: Identifier
Token: *, Type: Operator
Token: e, Type: Identifier
Token: ;, Type: Symbol
Syntax Analysis:
Syntax analysis completed successfully.
```

16. Invalid variable declaration

```
int main() {
    a % b;
}
```

```
Enter your C-like code (type 'END' on a new line to finish input):
int main() {
    a % b;
}
END
Input:
int main() {
    a % b;
}

Tokenized Output:
Token: int, Type: Keyword
Token: main, Type: Identifier
Token: (, Type: Symbol
Token: ), Type: Symbol
Token: {, Type: Symbol
Token: a, Type: Identifier
Token: b, Type: Identifier
Token: ;, Type: Symbol
Token: }, Type: Symbol
Syntax Analysis:
Syntax Error: Identifier 'a' found where it wasn't expected.

PS C:\Users\uditi\AppData\Local\Microsoft\Windows\INetCache\IE\Y637SSUO\output>
```

17. Ignoring multiline comments

```
int sum = a + 4; /* This is a multi-line comment
                    that should be ignored */
```

```
Enter your C-like code (type 'END' on a new line to finish input):
int sum = a + 4; /* This is a multi-line comment
                  that should be ignored */
END
Input:
int sum = a + 4; /* This is a multi-line comment
                  that should be ignored */

Tokenized Output:
Token: int, Type: Keyword
Token: sum, Type: Identifier
Token: =, Type: Operator
Token: a, Type: Identifier
Token: +, Type: Operator
Token: 4, Type: Number
Token: ;, Type: Symbol
Syntax Analysis:
Syntax analysis completed successfully.

PS C:\Users\uditi\AppData\Local\Microsoft\Windows\INetCache\IE\Y637SSUO\output>
```

18. Ignoring multiline comments

int sum = a + 4; // This is a comment

```
Enter your C-like code (type 'END' on a new line to finish input):
int sum = a + 4; // This is a comment
END
Input:
int sum = a + 4; // This is a comment

Tokenized Output:
Token: int, Type: Keyword
Token: sum, Type: Identifier
Token: =, Type: Operator
Token: a, Type: Identifier
Token: +, Type: Operator
Token: 4, Type: Number
Token: ;, Type: Symbol
Syntax Analysis:
Syntax analysis completed successfully.

PS C:\Users\uditi\AppData\Local\Microsoft\Windows\INetCache\IE\Y637SSUO\output> \
```