***TASK 1.(C++)***

#include <bits/stdc++.h>

using namespace std;

void isComment(string line)

{

if (line.size()>=2 && line[0] == '/' && line[1] == '/') {

cout << "It is a single-line comment";

return;

}

if (line.size()>=4 && line[line.size() - 2] == '\*'

&& line[line.size() - 1] == '/' && line[0] == '/' && line[1] == '\*') {

cout << "It is a multi-line comment";

return;

}

cout << "It is not a comment";

}

int main()

{

string line = "/\*Compiler Construction Task\*/";

isComment(line);

return 0;

}

A screenshot of a computer

Description automatically generated

***i) Describe the logic used.***

1.Check if at the first Index [0] the value is ‘/’ then follow below steps else print “It is not a comment”.

2.If line[0] == ‘/’:

* + If line[1] == ‘/’, then print “It is a single line comment”.
  + If line[1] == ‘\*’, then traverse the string and if any adjacent pair of ‘\*’ & ‘/’ is found then print “It is a multi-line comment”.

3.Otherwise, print “It is not a comment”.

***ii) Was lexical analysis or/and syntax analysis concepts important in your logic? If yes, how?***

-YES both concepts were used.

-lexical analysis would be responsible for recognizing comments (e.g., **//** and **/\* ... \*/**) as special tokens, separate from regular code tokens.

-syntax analysis would help distinguish between different language constructs, such as variable declarations, function definitions, control flow statements, and comments. It ensures that the code follows the syntactical rules of the C++ language.