**Name: Sparsh Karna**

**Reg Number: 23BDS1172**

**Lab Excercise: 2**

**Aim:**

The aim of this program is to remove all comments (single-line and multi-line) from a given source code file, and also normalize the whitespace.

**Procedure:**

1. To remove comments from a source code file, the program processes the input code character-by-character, tracking whether it is inside a string, a single-line comment, or a multi-line comment.
2. If it is inside a comment, characters are skipped until the comment ends.
3. If it is inside a string, all characters are preserved until the string closes.
4. After removing comments, the code is normalized by trimming extra spaces and reducing multiple blank lines into a single blank line.

**Algorithm:**

1. Read the entire source file content into a string.
2. Initialize flags for tracking whether parsing is inside a string, single-line comment, or multi-line comment.
3. Iterate through each character:
   1. Detect and skip comments based on their start and end markers.
   2. Preserve text within strings.
4. After removing comments, split the code into lines and trim spaces.
5. Remove consecutive blank lines while preserving single empty lines for readability.
6. Write the cleaned code to an output file.

**Program:**

**1. Code:**

#include <iostream>

#include <fstream>

#include <sstream>

#include <string>

using namespace std;

static inline bool isQuoteChar(char ch)

{

return ch == '"' || ch == '\'';

}

string stripComments23BDS1172(const string &src)

{

string out;

bool singleLine = false, multiLine = false, insideString = false;

char quoteType = 0;

for (size\_t idx = 0; idx < src.size(); ++idx)

{

char curr = src[idx];

char nxt = (idx + 1 < src.size()) ? src[idx + 1] : '\0';

if (!insideString && !singleLine && !multiLine && isQuoteChar(curr))

{

insideString = true;

quoteType = curr;

out.push\_back(curr);

}

else if (insideString && curr == quoteType && src[idx - 1] != '\\')

{

insideString = false;

out.push\_back(curr);

}

else if (!insideString && !singleLine && !multiLine && curr == '/' && nxt == '/')

{

singleLine = true;

++idx;

}

else if (!insideString && !singleLine && !multiLine && curr == '/' && nxt == '\*')

{

multiLine = true;

++idx;

}

else if (singleLine && curr == '\n')

{

singleLine = false;

out.push\_back('\n');

}

else if (multiLine && curr == '\*' && nxt == '/')

{

multiLine = false;

++idx;

}

else if (!singleLine && !multiLine)

{

out.push\_back(curr);

}

}

return out;

}

string removeExtraSpaces(const string &txt)

{

istringstream reader(txt);

ostringstream writer;

string prevNonEmpty, current;

while (getline(reader, current))

{

size\_t first = current.find\_first\_not\_of(" \t\r\n");

size\_t last = current.find\_last\_not\_of(" \t\r\n");

string cleaned = (first == string::npos) ? "" : current.substr(first, last - first + 1);

if (cleaned.empty() && prevNonEmpty.empty())

continue;

writer << cleaned << '\n';

prevNonEmpty = cleaned;

}

return writer.str();

}

int main()

{

const string sourceFile = "input.txt";

const string destFile = "output.txt";

ifstream check(sourceFile);

if (!check.is\_open())

{

cerr << "Cannot locate " << sourceFile << "!\n";

return 1;

}

stringstream srcBuffer;

srcBuffer << check.rdbuf();

string noComments = stripComments23BDS1172(srcBuffer.str());

string neatCode = removeExtraSpaces(noComments);

ofstream output(destFile);

output << neatCode;

cout << "Created file: " << destFile << endl;

return 0;

}

**2. input.txt:**

#include // include vector library #include /\* include string library \*/

int sum(int a, int b) { // function to sum two numbers return a + b; /\* return result \*/ }

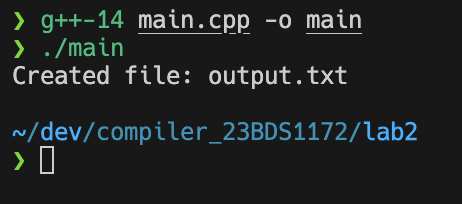
int main() { // variable declarations int x23BDS1172 = 5; /\* first number \*/ int y = 7; // second number

/\*  
 Adding two numbers and  
 printing the result  
\*/  
int result = sum(x, y);  
cout << "Sum is: " << result << endl; // output result  
return 0;

}

**Output:**

**Terminal Output:**



**Output.txt:**

**#include <vector>**

**#include <string>**

**int sum(int a, int b) {**

**return a + b;**

**}**

**int main() {**

**int x23BDS1172 = 5;**

**int y = 7;**

**int result = sum(x, y);**

**cout << "Sum is: " << result << endl;**

**return 0;**

**}**

**Result:**

The program successfully removed all comments from the given code while preserving code inside strings and normalizing whitespace.