**Name** – Bhargav Shamuvel Gurav **PRN** – 2041009

**Class** – L.Y. B-Tech (Computer) **Batch** – B1

**Course Code** – CO406U **Course Name** - CDL

**Practical no. A (2)**

**Aim:** Write a program to identify whether a given line is a comment or not.

**Theory :**

If you want to identify comments in a C program, you'll need to consider C's comment syntax. In C, there are two main types of comments: single-line comments starting with `//` and multi-line comments enclosed within `/\*` and `\*/`. You can write a program that detects both types of comments. Here's an example in Python that identifies comments in a C program:

def is\_single\_line\_comment(line):

line = line.strip()

return line.startswith('//')

def is\_multi\_line\_comment\_start(line):

line = line.strip()

return line.startswith('/\*')

def is\_multi\_line\_comment\_end(line):

line = line.strip()

return line.endswith('\*/')

def is\_comment(line):

return is\_single\_line\_comment(line) or is\_multi\_line\_comment\_start(line)

def detect\_comment\_type(line, inside\_multi\_line\_comment):

if inside\_multi\_line\_comment:

if is\_multi\_line\_comment\_end(line):

return False

return True

if is\_single\_line\_comment(line) or is\_multi\_line\_comment\_start(line):

return True

return False

inside\_multi\_line\_comment = False

with open("c\_program.c", "r") as file:

for line in file:

line = line.strip()

inside\_multi\_line\_comment=detect\_comment\_type(line, inside\_multi\_line\_comment)

if inside\_multi\_line\_comment:

print("Comment:", line)

else:

print("Code:", line)

In this Python program:

1. We define four functions: `is\_single\_line\_comment`, `is\_multi\_line\_comment\_start`, `is\_multi\_line\_comment\_end`, and `is\_comment`. These functions check if a given line is a single-line comment, the start of a multi-line comment, the end of a multi-line comment, or any type of comment, respectively.

2. The `detect\_comment\_type` function is responsible for determining whether a line is part of a multi-line comment. It checks if the line starts or ends a multi-line comment or if it is inside a multi-line comment.

3. We read the C program line by line from a file and use the `detect\_comment\_type` function to track whether we are inside a multi-line comment or not. Based on this information, we print the lines as either code or comments.

Please make sure to replace `"c\_program.c"` with the actual path to your C program file. This program should correctly identify both single-line and multi-line

comments in a C program.

**Program Code:**

#include<stdio.h>

#include<cstring>

int countSingle = 0;

int countMulti = 0;

int multiFlag = 0;

int lineCount = 0;

void commentsCheck(char line[] ,int lineCount);

int linePrint(char line[], int i)

{

int x;

for(x=i;x<strlen(line);x++)

{

printf("%c",line[x]);

}

return x;

}

int linePrintMulti(char line[], int i)

{

int x=i;

while(x<strlen(line))

{

if(line[x]=='\*'&&line[x+1]=='/')

{

multiFlag = 0;

return x;

}

printf("%c",line[x]);

x++;

}

multiFlag = 1;

return x;

}

int main(int argc, char\* argv[])

{

//read any text file from currect directory

char const\* const fileName = "comments.c";

FILE\* file = fopen(fileName, "r");

if(!file)

{

printf("\n Unable to open : %s ", fileName);

return -1;

}

char line[500];

while (fgets(line, sizeof(line), file))

{

lineCount++;

commentsCheck(line , lineCount);

}

fclose(file);

printf("\nRESULT:\n");

printf("\n Single line Comments: %d ", countSingle);

printf("\n Multiple line Comments: %d ", countMulti);

return 0;

}

void commentsCheck(char line[] , int linec)

{

int i = 0;

// Single Line Comment

while(i<strlen(line) || multiFlag == 1)

{

if(line[i] == '/' && line[i+1] == '/')

{

printf("\n\nIt is a single-line comment at %d\n",linec);

countSingle++;

i = linePrint(line, i+2);

}

else if(line[i] == '/' && line[i+1] == '\*')

{

printf("\n\nIt is a multi-line comment at %d\n" , linec);

multiFlag = 1;

countMulti++;

i = linePrintMulti(line,i+2);

}

else if(multiFlag==1)

{

i = linePrintMulti(line,i);

return;

}

i++;

}

return;

}

**Input Code (comments.c):**

// CDL

// this is a comment

#include <stdio.h>

/\* way of single line comment \*/

/\*

Multiline comment

\*/

int main() {

printf("Hello, World!");

return 0;

}

**Output:**

****

**Conclusion :** In this practical we learnt how the program identifies if there is a comment in input code.