**Programs**

**[4] Check Whether Given String Belongs To CFG : S → 0S0 | 1S1 | 0 | 1 | ε**

#include <stdio.h>

#include <string.h>

int belongsToCFG(char \*str, int start, int end)

{

if (start > end)

return 1;

if (str[start] == '0' && str[end] == '0')

return belongsToCFG(str, start + 1, end - 1);

else if (str[start] == '1' && str[end] == '1')

return belongsToCFG(str, start + 1, end - 1);

else if (str[start] == '0')

return belongsToCFG(str, start + 1, end);

else if (str[start] == '1')

return belongsToCFG(str, start + 1, end);

return 0;

}

int main()

{

char str[100];

printf("Enter a String : ");

scanf("%s", str);

int len = strlen(str);

if (belongsToCFG(str, 0, len - 1))

printf("String Belongs to Context Free Grammar.\n");

else

printf("String Doesn't Belong to Context Free Grammar.\n");

return 0;

}

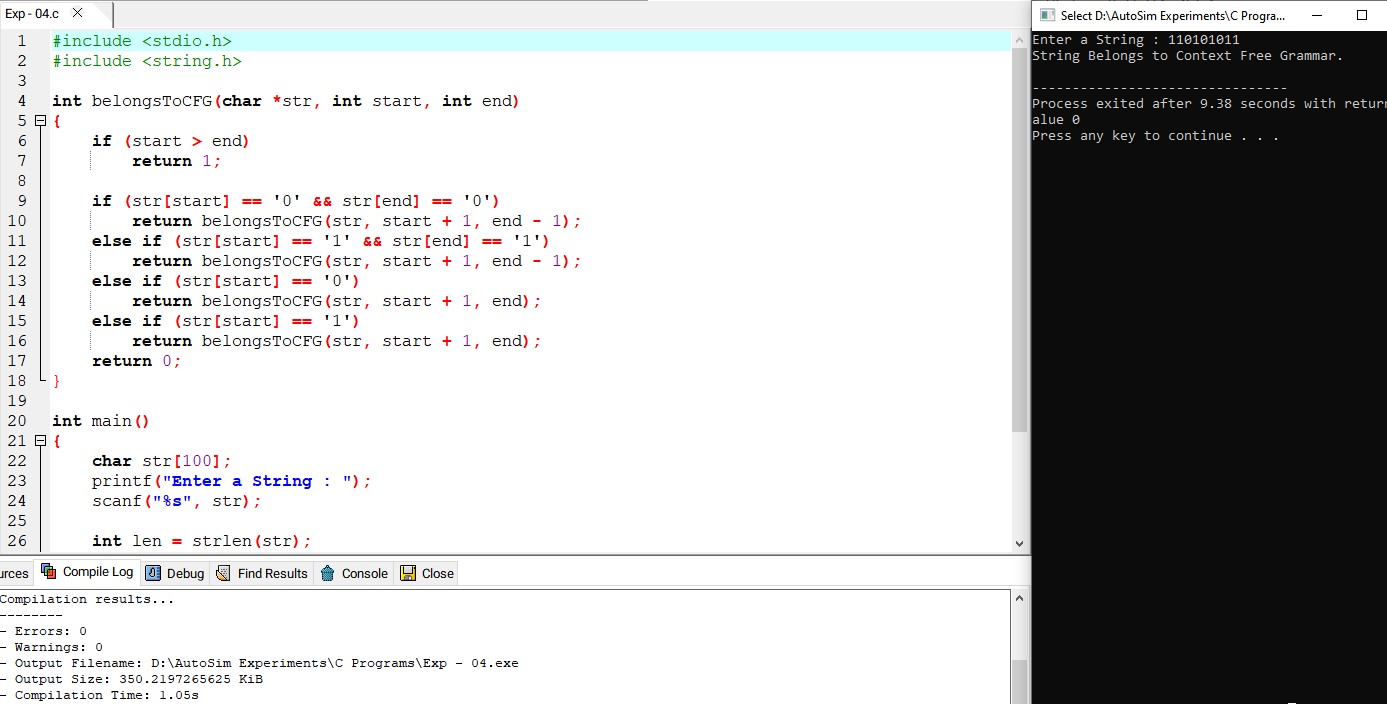
**Algorithm**

**Initialization :** Create a table where each cell [i, j] represents a substring from position i to position j in the input string.

**Fill Terminals :** Populate cells [i, i] with non-terminals that generate the corresponding input character at position i.

**Fill Non-Terminals :** For each substring length len from 2 to input length, iterate over i and partition substring into [i, i+k] and [i+k+1, i+len]. Check non-terminals if they generate these substrings, then add them to cell [i, i+len].

**Acceptance :** If start symbol exists in cell [0, n-1], where n is input length, the input string belongs to the CFG.



**Result**

Thus we have successfully implemented and executed the program and the strings given as inputs are verified