3.Write a C program to check whether a given string belongs to the language defined by a Context Free Grammar (CFG)

S → 0A1 A → 0A | 1A | ε

AIM: To Write a C program to check whether a given string belongs to the language defined by a Context Free Grammar (CFG)

S → 0A1 A → 0A | 1A | ε

ALGORITHM:

1.Define the CFG:

Define the Context-Free Grammar (CFG) with its productions

S → 0A1

A → 0A | 1A | ε

2.Implement Parsing Functions:

Implement parsing functions for each non-terminal in the CFG. In this case, you'll have functions for parse\_S and parse\_A. Each function should consume part of the input string according to the CFG rules.

3.Parsing S Production:

In the parse\_S function, start with the production S → 0A1. Check if the input string starts with '0'. If it does, consume '0', call parse\_A, and then check for '1'. If all these steps succeed, return the remaining input string.

4.Parsing A Production:

In the parse A function, implement the three production rules for A → 0A, A → 1A, and A → ε. Recursively call parse\_A based on the rules and consume the input string accordingly.

5.Check for Successful Parsing:

Create a function, let's call it is\_in\_language, which calls parse\_S with the input string. If parse\_S returns a non-None value (i.e., the entire input string is successfully parsed), then the input string belongs to the language.

6.Example Usage:

Test the algorithm with different input strings using the is\_in\_language function.

7.Repeat Testing:

Test the algorithm with various input strings to ensure its correctness

PROGRAM:  
#include <stdio.h>

#include <string.h>

int isBelongToCFG(char \*str);

int parseString(char \*str, int index);

int main() {

char input[100];

printf("Enter a string: ");

scanf("%s", input);

if (isBelongToCFG(input)) {

printf("The string belongs to the language defined by the CFG.\n");

} else {

printf("The string does not belong to the language defined by the CFG.\n");

}

return 0;

}

int isBelongToCFG(char \*str) {

return parseString(str, 0);

}

int parseString(char \*str, int index) {

if (str[index] == '0') {

index++;

if (str[index] == 'A') {

index++;

if (str[index] == '1') {

index++;

return parseString(str, index);

}

}

}

else if (str[index] == '1') {

index++;

if (str[index] == 'A') {

index++;

return parseString(str, index);

}

} else if (str[index] == 'A') {

index++;

return parseString(str, index);

}

if (str[index] == '\0') {

return 1;

} else {

return 0;

}

}

OUTPUT:

