

EXPERIMENT-1

AIM:

Write a program to check whether a given string belongs to a grammar or not.

PROGRAMS:

1. Grammar: $S \rightarrow aS$, $S \rightarrow Sb$, $S \rightarrow ab$

```
#include <bits/stdc++.h>
using namespace std;
int main() {
    string str;
    bool flag = true;
    cout << "The grammar is:  $S \rightarrow aS$ ,  $S \rightarrow Sb$ ,  $S \rightarrow ab$ " << endl;
    cout << "Enter the string to be checked: ";
    cin >> str;
    int n = str.length();
    if (str[0] == 'a' && str[n - 1] == 'b'){
        for (int i = 1; i < str.length(); i++){
            if (str[i] == 'b'){
                flag = false;
            }
            else if (str[i] == 'a' && flag == false){
                cout << "String is not accepted";
                exit(0);
            }
        }
        cout << "String is accepted";
    }
    else{
        cout << "String is not accepted";
    }
    return 0;
}
```

OUTPUT:

```
The grammar is:  $S \rightarrow aS$ ,  $S \rightarrow Sb$ ,  $S \rightarrow ab$ 
Enter the string to be checked: aaabb
String is accepted
```

2. Grammar: $S \rightarrow aSa$, $S \rightarrow bSb$, $S \rightarrow a$, $S \rightarrow b$

```

#include <bits/stdc++.h>
using namespace std;
int main() {
    string str;
    bool flag = true;
    cout << "The grammar is: S->aSa, S->bSb, S->a, S->b" << endl;
    cout << "Enter the string to be checked: ";
    cin >> str;
    int n = str.length();
    int a = 0, b = n - 1;
    if (n % 2 != 0) {
        while (b > a) {
            if (str[a] == str[b]) {
                a++;
                b--;
            }
            else {
                cout << "String is not accepted";
                exit(0);
            }
        }
        cout << "String is accepted";
    }
    else {
        cout << "String is not accepted";
    }
    return 0;
}

```

OUTPUT:

```

The grammar is: S->aSa, S->bSb, S->a, S->b
Enter the string to be checked: abbabba
String is accepted

```

3. Grammar: $S \rightarrow aSbb$, $S \rightarrow abb$

```

#include <iostream>
using namespace std;
int main() {
    string str;
    bool flag = true;
    int a_count = 0, b_count = 0;
    cout << "The grammar is: S->aSbb, S->abb" << endl;
    cout << "Enter the string to be checked: ";
    cin >> str;
    int n = str.length();
    if (str[0] == 'a' && str[n - 1] == 'b'){
        for (int i = 0; i < str.length(); i++){
            if (str[i] == 'a' && flag == false){
                cout << "String is not accepted";
                exit(0);
            }
            else if (str[i] == 'a' && flag == true){
                a_count++;
            }
            else if (str[i] == 'b'){
                b_count++;
                flag = false;
            }
        }
        if (b_count == 2 * a_count){
            cout << "String is accepted";
        }
        else {
            cout << "String is not accepted";
        }
    }
    else {
        cout << "String is not accepted";
    }
    return 0;
}

```

OUTPUT:

```

The grammar is: S->aSbb, S->abb
Enter the string to be checked: aaabbbbbbb
String is accepted

```

4. Grammar: $S \rightarrow aSb$, $S \rightarrow ab$

```

#include <iostream>
using namespace std;
int main() {
    string str;
    bool flag = true;
    int a_count = 0, b_count = 0;
    cout << "The grammar is: S->aSb, S->ab" << endl;
    cout << "Enter the string to be checked: ";
    cin >> str;
    int n = str.length();
    if (str[0] == 'a' && str[n - 1] == 'b'){
        for (int i = 0; i < str.length(); i++){
            if (str[i] == 'a' && flag == false){
                cout << "String is not accepted";
                exit(0);
            }
            else if (str[i] == 'a' && flag == true){
                a_count++;
            }
            else if (str[i] == 'b'){
                b_count++;
                flag = false;
            }
        }
    }
    if (b_count == a_count){
        cout << "String is accepted";
    }
    else{
        cout << "String is not accepted";
    }
    else{
        cout << "String is not accepted";
    }
    return 0;
}

```

OUTPUT:

```

The grammar is: S->aSb, S->ab
Enter the string to be checked: aaaabbbb
String is accepted

```

VIVA-VOCE QUESTIONS:

Ques 1. What is the key feature of a CFG?

Ans 1. CFGs use production rules to generate strings in a language.

Ques 2. How do you determine if a string is in a CFG's language?

Ans 2. By constructing a parse tree for the string.

Ques 3. What are terminal symbols in a CFG?

Ans 3. Symbols that appear in the input string.

Ques 4. What is the significance of the Pumping Lemma for CFGs?

Ans 4. It helps identify non-context-free languages.

Ques 5. Are all programming languages context-free?

Ans 5. No, many programming languages have context-sensitive syntax.