# Premier University Chittagong

| | | |
|---|---|---|
| Course Title | : | Compiler Construction Laboratory |
| Course Code | : | CSE 454 |
| Department | : | Computer Science and Engineering |
| Report No | : | 03 |
| Report Name | : | Write a program for the derivation of a string for context free grammar |
| Date of Submission | : | 23/08/2020 |

Marks

| | |
|---|---|
| Name | : Joyanta Dutta |
| ID | : 1402710200740 |
| Sec | : C8A |
| Batch | : 27th |

**Objective:** Write a program for the derivation of a string for context free grammar.

**Introduction**:

Context free grammar: A context-free grammar is a set of recursive rules used to generate patterns of strings. A context-free grammar can describe all regular languages and more, but they cannot describe all possible languages. Context-free grammars are studied in fields of theoretical computer science, compiler design, and linguistics.

Derivation: A derivation of a string for a grammar is a sequence of grammar rule applications that transform the start symbol into the string. A derivation proves that the string belongs to the grammar's language. A derivation is fully determined by giving, for each step: the rule applied in that step.

**Software Requirements:**

1. ASUS k43u Laptop
2. 2 GB Ram and 512 GB hard disk.
3. CodeBlocks
4. GNU GCC compiler

**Description:** Basically this code is checking the result of the left most derivation and the right most derivation of a string by using the context free grammar.

For the first sample input we can get,

Left most derivation:

S → AB → aaAB → aaB → aaBb → aab

Right most derivation:

S → AB → ABb → Bb → aaAb → aab

For the second sample input we can get,

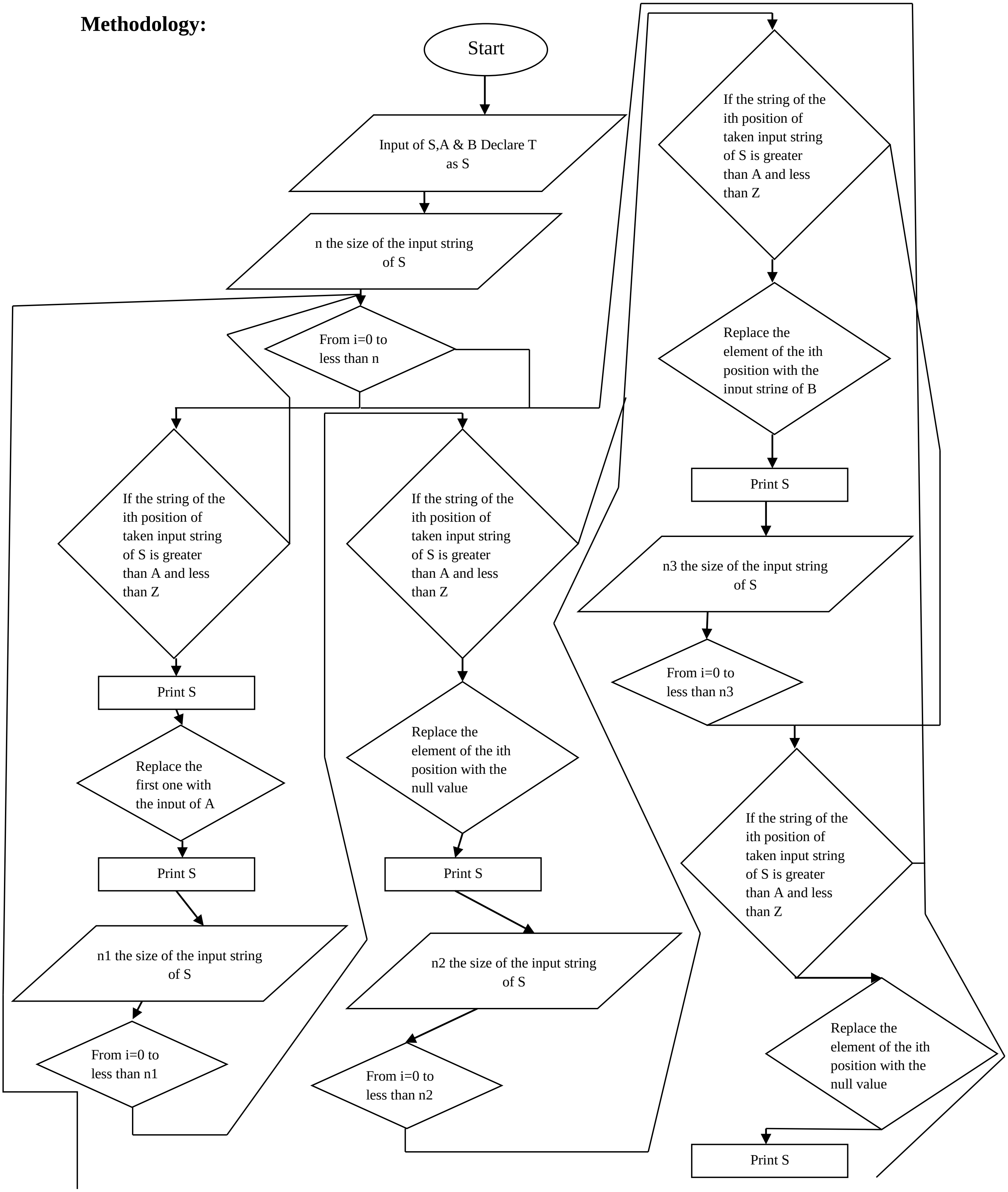
Left most derivation:

S → CD → ccCD → ccD → ccDd → ccd

Right most derivation:

S → CD → CDd → Dd → ccCd → ccd

## Methodology:


Start
Input of S,A & B Declare T as S
n the size of the input string of S
From i=0 to less than n
If the string of the ith position of taken input string of S is greater than A and less than Z
Print S
Replace the first one with the input of A
Print S
n1 the size of the input string of S
From i=0 to less than n1
If the string of the ith position of taken input string of S is greater than A and less than Z
Replace the element of the ith position with the null value
Print S
n2 the size of the input string of S
From i=0 to less than n2
If the string of the ith position of taken input string of S is greater than A and less than Z
Replace the element of the ith position with the input string of B
Print S
n3 the size of the input string of S
From i=0 to less than n3
If the string of the ith position of taken input string of S is greater than A and less than Z
Replace the element of the ith position with the null value
Print S

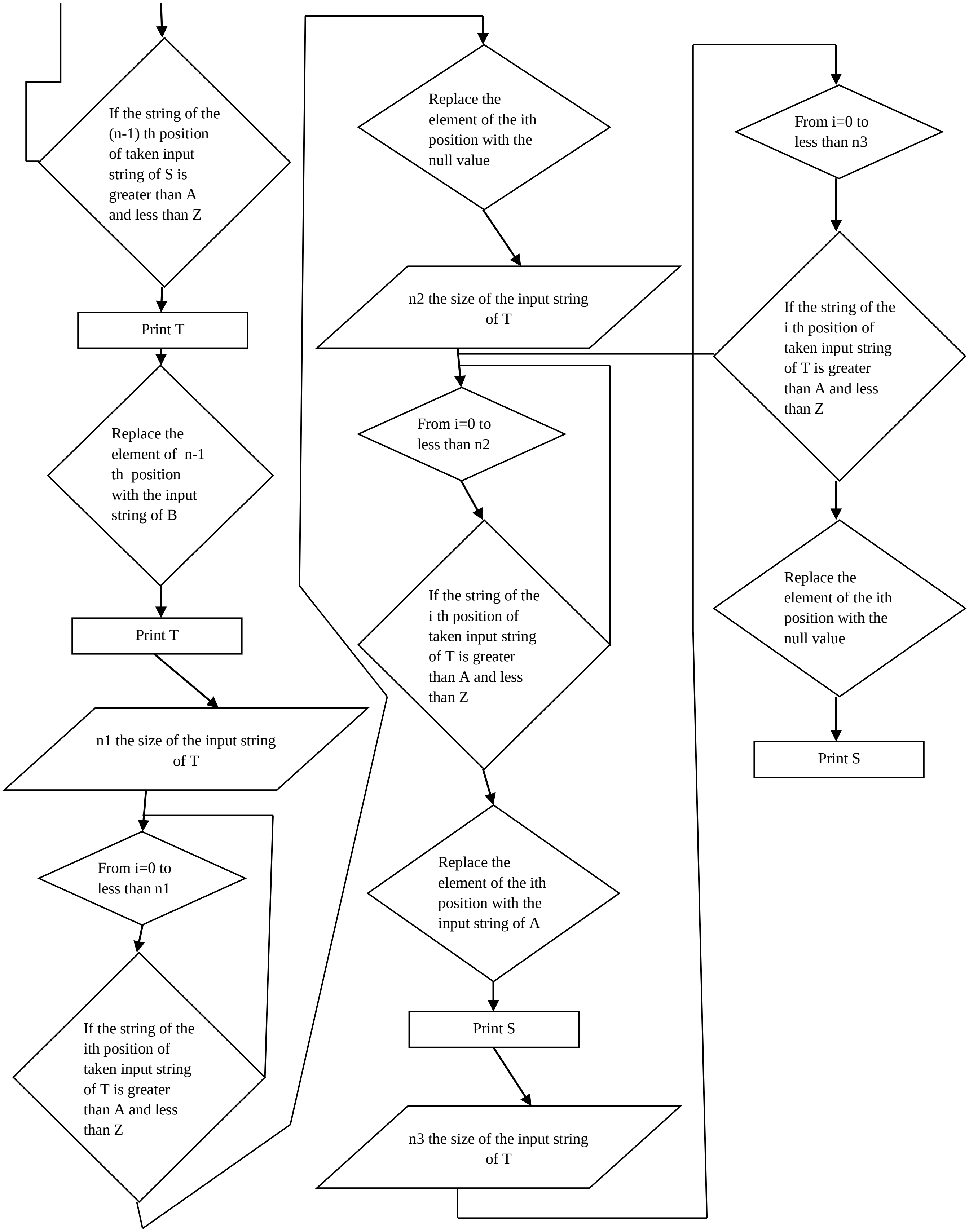

If the string of the (n-1) th position of taken input string of S is greater than A and less than Z
Print T
Replace the element of n-1 th position with the input string of B
Print T
n1 the size of the input string of T
From i=0 to less than n1
If the string of the ith position of taken input string of T is greater than A and less than Z
Replace the element of the ith position with the null value
n2 the size of the input string of T
From i=0 to less than n2
If the string of the i th position of taken input string of T is greater than A and less than Z
Replace the element of the ith position with the input string of A
Print S
n3 the size of the input string of T
From i=0 to less than n3
If the string of the i th position of taken input string of T is greater than A and less than Z
Replace the element of the ith position with the null value
Print S

Fig 1**:** Flow process of a program for the derivation of a string for context free grammar

**Source Code:**

```
#include<bits/stdc++.h>

using namespace std;

int main(){
string S;
string T;
string A;
string B;
int i,n,n1,n2,n3;

cout<<"Get the value of S: ";
cin>>S;
cout<<"Get the value of the first one: ";
cin>>A;
cout<<"Get the value of the second one: ";
cin>>B;

T=S;

n = S.length();
```

```
for(i=0;i<n;i++){
    if(S[i]>='A'&&S[i]<='Z'){
            cout<<"\n";
            cout<<"Result of the leftmost derivation"<<endl;
            cout<<"S -> "<<S<<endl;
        S.replace(0,1,A);
        cout<<"S -> "<<S<<endl;
        n1 = S.length();

        for(i=0;i<n1;i++){
    if(S[i]>='A'&&S[i]<='Z'){
        S.replace(i,1,"");
        cout<<"S -> "<<S<<endl;
        }
        }

        n2 = S.length();

        for(i=0;i<n2;i++){
    if(S[i]>='A'&&S[i]<='Z'){
        S.replace(i,1,B);
        cout<<"S -> "<<S<<endl;
        }
        }
        n3 = S.length();

        for(i=0;i<n3;i++){
    if(S[i]>='A'&&S[i]<='Z'){
        S.replace(i,1,"");
        cout<<"S -> "<<S<<endl;
        }
        }
    }

    if(T[n-1]>='A'&&T[n-1]<='Z'){
            cout<<"\n";
            cout<<"Result of the rightmost derivation"<<endl;
            cout<<"S -> "<<T<<endl;
        T.replace(n-1,1,B);
        cout<<"S -> "<<T<<endl;
        n1 = T.length();

        for(i=0;i<n1;i++){
    if(T[i]>='A'&&T[i]<='Z'){
        T.replace(i,1,"");
        cout<<"S -> "<<T<<endl;
        }
        }
```

```
        n2 = T.length();


        for(i=0;i<n2;i++){
    if(T[i]>='A'&&T[i]<='Z'){
        T.replace(i,1,A);
        cout<<"S -> "<<T<<endl;
        }
        }


        n3 = T.length();


        for(i=0;i<n3;i++){
    if(T[i]>='A'&&T[i]<='Z'){
        T.replace(i,1,"");
        cout<<"S -> "<<T<<endl;
        }
        }
    }
    else{
        cout<<"Can't fulfill the requirement";
        break;
    }
}
}
```

**Input:**

Sample 1:

Get the value of S: AB

Get the value of the first one: aaA

Get the value of the second one: Bb

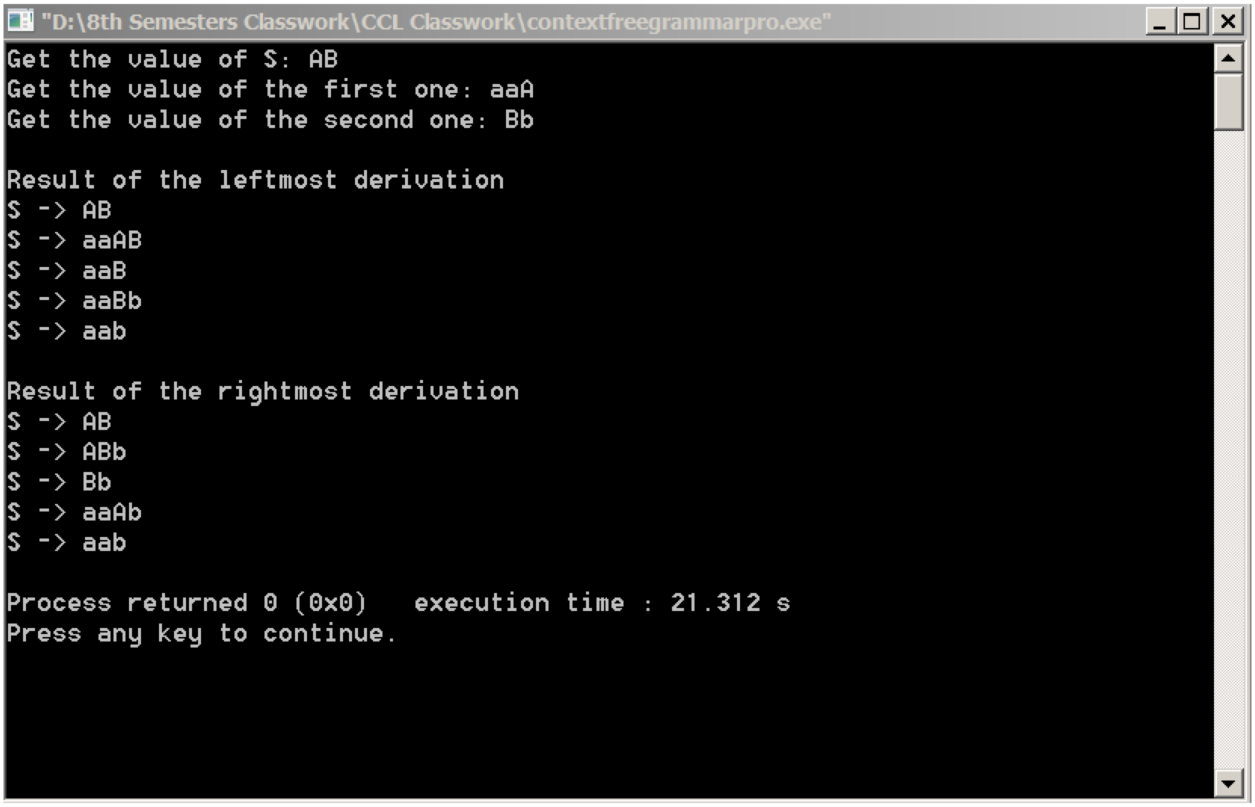
Sample 2:

Get the value of S: CD

Get the value of the first one: ccC

Get the value of the second one: Dd

**Output:**



"D:\8th Semesters Classwork\CCL Classwork\contextfreegrammarpro.exe"

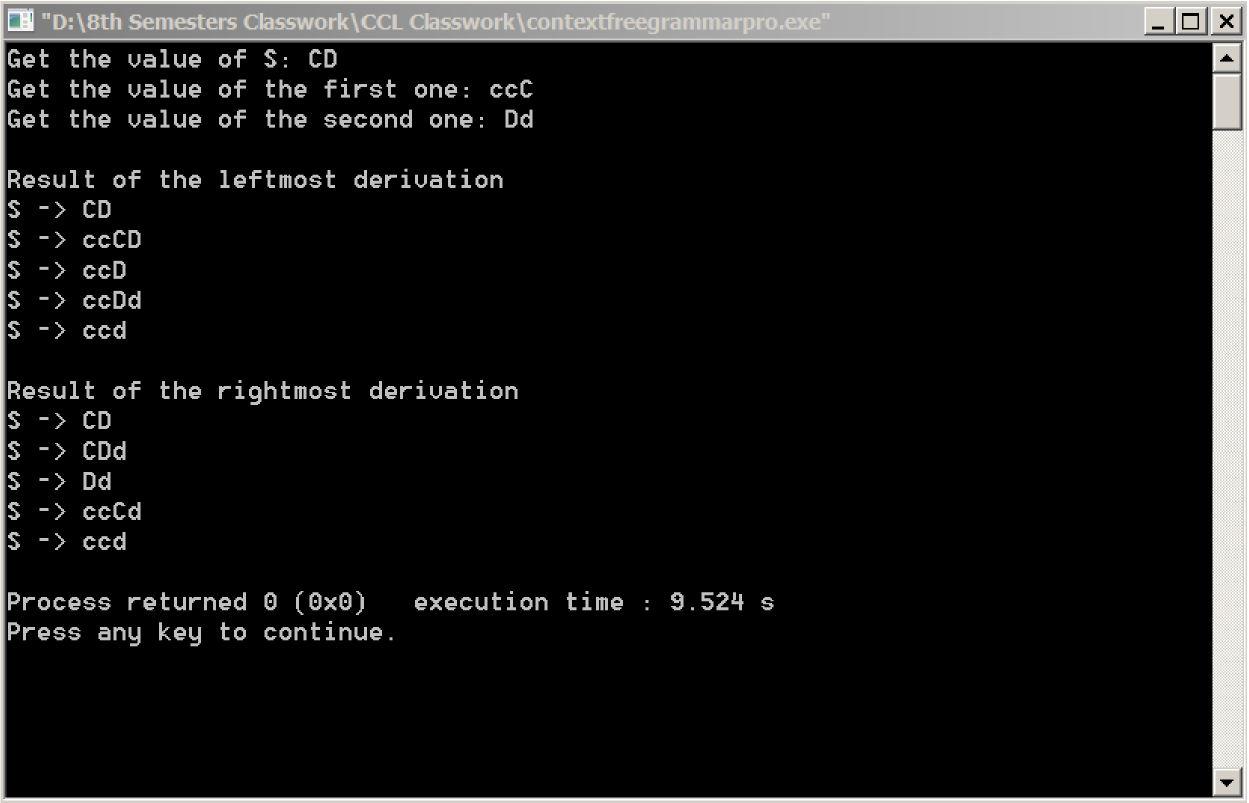
```
Get the value of S: AB
Get the value of the first one: aaA
Get the value of the second one: Bb

Result of the leftmost derivation
S -> AB
S -> aaAB
S -> aaB
S -> aaBb
S -> aab

Result of the rightmost derivation
S -> AB
S -> ABb
S -> Bb
S -> aaAb
S -> aab

Process returned 0 (0x0)   execution time : 21.312 s
Press any key to continue.
```

Fig 2: Output of sample input 1


"D:\8th Semesters Classwork\CCL Classwork\contextfreegrammarpro.exe"

```
Get the value of S: CD
Get the value of the first one: ccC
Get the value of the second one: Dd

Result of the leftmost derivation
S -> CD
S -> ccCD
S -> ccD
S -> ccDd
S -> ccd

Result of the rightmost derivation
S -> CD
S -> CDd
S -> Dd
S -> ccCd
S -> ccd

Process returned 0 (0x0)   execution time : 9.524 s
Press any key to continue.
```

Fig 3: Output of sample input 2

**Discussion:** This code is built for the derivation of a string for context free grammar. basically its checking the left most derivation and the right most derivation of string for context free grammar. For the first sample input we can get,

Left most derivation:

S → AB→ aaAB → aaB → aaBb→ aab

Right most derivation:

S → AB→ ABb→Bb → aaAb→aab

For the second sample input we can get,

Left most derivation:

S → CD→ ccCD→ ccD →ccDd→ ccd

Right most derivation:

S → CD→ CDd→Dd→ ccCd→ccd