

# Dept. of Computer Science & Engineering



আন্তর্জাতিক ইসলামী বিশ্ববিদ্যালয় চট্টগ্রাম  
الجامعة الإسلامية العالمية شيتا خونج  
International Islamic University Chittagong

**Course Code:** CSE-3528

**Course Title:** Compiler Lab

## Spring-22

**Submitted By:**

**Name:** Farida Nusrat

**ID:** C201242

**Semester:** 5<sup>th</sup>

**Section:** 5AF

**Email:** [c201242@ugrad.iiuc.ac.bd](mailto:c201242@ugrad.iiuc.ac.bd)

**Submitted To:**

Mrs. Israt Binteh Habib

Dept. Of CSE, IIUC.

**Signature:**

Farida Nusrat

**Submission Date:** 25.11.22

## **OBJECTIVE:**

C Program for Implementation of Predictive Parser.

## **RESOURCE:**

Code-blocks

## **PROGRAM LOGIC:**

- Read the input string.
- By using the FIRST AND FOLLOW values.
- Verify the FIRST of non-terminal and insert the production in the FIRST value
- If we have any @ terms in FIRST then insert the productions in FOLLOW values
- Constructing the predictive parser table.

## **PROCEDURE:**

Go to debug -> run or press CTRL + F9 to run the program.

## **PROGRAM:**

```
#include <bits/stdc++.h>
using namespace std;

char pro[7][10] = {"S", "A", "A", "B", "B", "C", "C"};
char pr[7][10] = {"A", "Bb", "Cd", "aB", "@", "Cc", "@"};
char prod[7][10] = {"S->A", "A->Bb", "A->Cd", "B->aB", "B->@", "C->Cc", "C->@"};
char first[7][10] = {"abcd", "ab", "cd", "a@", "@", "c@", "@"};
char follow[7][10] = {"$", "$", "$", "a$", "b$", "c$", "d$"};
char table[5][6][10];
int numr(char c)
{
    switch (c)
    {
        case 'S':
            return 0;
        case 'A':
            return 1;
        case 'B':
            return 2;
        case 'C':
            return 3;
        case 'a':
            return 0;
        case 'b':
            return 1;
        case 'c':
```

```

        return 2;
    case 'd':
        return 3;
    case '$':
        return 4;
    }
    return (2);
}
int main()
{
    int i, j, k;
    for (i = 0; i < 5; i++)
        for (j = 0; j < 6; j++)
            strcpy(table[i][j], " ");
    cout << "\nThe following is the predictive parsing table for the following grammar:\n";
    for (i = 0; i < 7; i++)
        cout << prod[i] << "\n";
    cout << "\nPredictive parsing table is\n";
    fflush(stdin);
    for (i = 0; i < 7; i++)
    {
        k = strlen(first[i]);
        for (j = 0; j < 10; j++)
            if (first[i][j] != '@')
                strcpy(table[numr(prol[i][0]) + 1][numr(first[i][j]) + 1], prod[i]);
    }
    for (i = 0; i < 7; i++)
    {
        if (strlen(pror[i]) == 1)
        {
            if (pror[i][0] == '@')
            {
                k = strlen(follow[i]);
                for (j = 0; j < k; j++)
                    strcpy(table[numr(prol[i][0]) + 1][numr(follow[i][j]) + 1], prod[i]);
            }
        }
    }
    strcpy(table[0][0], " ");
    strcpy(table[0][1], "a");
    strcpy(table[0][2], "b");
    strcpy(table[0][3], "c");
    strcpy(table[0][4], "d");
    strcpy(table[0][5], "$");
    strcpy(table[1][0], "S");
    strcpy(table[2][0], "A");

```

```

strcpy(table[3][0], "B");
strcpy(table[4][0], "C");
printf("\n-----\n");
for (i = 0; i < 5; i++)
    for (j = 0; j < 6; j++)
    {
        cout << table[i][j];
        if (j == 5)
            cout << "\n-----\n";
    }
return 0;
}

```

### INPUT & OUTPUT:

```

input
The following is the predictive parsing table for the following grammar:
S->A
A->Bb
A->Cd
B->aB
B->@
C->Cc
C->@

Predictive parsing table is

-----
ab cd $
-----
SS->AS->AS->AS->A
-----
AA->BbA->BbA->CdA->Cd
-----
BB->aBB->@B->@ B->@
-----
C C->@C->@C->@
-----

...Program finished with exit code 0
Press ENTER to exit console.

```

### OUTCOMES:

The program is successfully written and created in the C++ language.