Practical - 2

2CS701 – Compiler Construction

Harshit Gajipara
19BCE059

Aim:

To Implement a Recursive Descent Parser Algorithm for the Grammar.

Code:

```
/*
To Implement a Recursive Descent Parser Algorithm for the
Grammer.

E --> T + E | T - E | T
T --> F * T | F / T | F
F --> ID | NUM | (E)
*/

#include <stdio.h>
#include <string.h>
#include <ctype.h>

char input[20];
int i, error;

void E();
void T();
void F();

void main()
{
    i = 0;
    error = 0;

    printf("\nEnter an arithmetic expression : ");
    gets(input);
```

```
    printf("\nInput\tAction\n------------------------------
-\n");
    E();

    if (strlen(input) == i && error == 0)
    {
        printf("\n---------------------------------\n");
        printf("\nString is successfully parsed!\n\n");
    }
    else
    {
        printf("\n---------------------------------\n");
        printf("Error in parsing String\n\n");
    }
}

void E()
{
    T();
    if (input[i] == '+' || input[i] == '-')
    {
        printf("%c\tE->T%cE \n", input[i], input[i]);
        i++;
        E();
    }
}

void T()
{
    F();
    if (input[i] == '*' || input[i] == '/')
    {
        printf("%c\tT->F%cT \n", input[i], input[i]);
        i++;
        T();
    }
}
```

```
void F()
{
    if (isalpha(input[i]))
    {
        printf("%c\tF->ID \n", input[i]);
        i++;
    }
    else if (isalnum(input[i]))
    {
        printf("%c\tF->NUMBER \n", input[i]);
        i++;
    }
    else if (input[i] == '(')
    {
        printf("%c\tF->(E) \n", input[i]);
        i++;
        E();
        if (input[i] == ')')
        {
            printf("%c\tF->(E) \n", input[i]);
            i++;
        }
        else
            error = 1;
    }
    else
        error = 1;
}
```

Output:

```
PS C:\Users\HARSHIT> cd "d:\19BCE059\B.Tech Semester 7\CC\C

Enter an arithmetic expression : ((8/2)-3)*6-(2*4)

Input   Action
--------------------------------
(       F->(E)
(       F->(E)
8       F->NUMBER
/       T->F/T
2       F->NUMBER
)       F->(E)
-       E->T-E
3       F->NUMBER
)       F->(E)
*       T->F*T
6       F->NUMBER
-       E->T-E
(       F->(E)
2       F->NUMBER
*       T->F*T
4       F->NUMBER
)       F->(E)

--------------------------------

String is successfully parsed!
```

```
PS D:\19BCE059\B.Tech Semester 7\CC\CC Practi

Enter an arithmetic expression : (2//3)

Input   Action
--------------------------------
(       F->(E)
2       F->NUMBER
/       T->F/T
/       T->F/T
3       F->NUMBER
)       F->(E)

--------------------------------
Error in parsing String
```

Conclusion:

In this practical, we learnt that using RDP we can parse any input for given grammar and check if the input is accepted by grammar or not.