SSN COLLEGE OF ENGINEERING, KALAVAKKAM

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

UCS1602 - Compiler Design Programming

Assignment-4

Implementation of Recursive Descent Parser

Name: Rahul Ram M

Class: CSE - B

Reg No: 185001121

Date: 4/03/2021

CODE:

ex_04.c

```
#include<stdio.h>

#include<string.h>

#include<ctype.h>

char productions[10][5];

int index = 0, size = 0, error = 0;

void Eprime();

void T();

void Tprime();

void F();

void E()

{

if(error == 0)
```

```
{
printf("T() ");
}
printf("E() ");
T();
Eprime();
}
void Eprime()
{
if(error == 0)
{
printf("T() ");
}
printf("E'() ");
if(strcmp(productions[index],"+") == 0)
{
index++;
T();
Eprime();
}
}
void T()
{
if(error == 0)
```

```
{
printf("T() ");
}

F();
Tprime();
}
void Tprime()
{
if(error == 0)
{
printf("T'() ");
}

if(strcmp(productions[index], "*") == 0)
{
index++;
F();
Tprime();
}
}
void F()
{
if(error == 0)
```

```
{
printf("F() ");
}

if(strcmp(productions[index], "id") == 0)
{
index++;
}
else if(strcmp(productions[index], "(") == 0)
{
index++;
E();
if(strcmp(productions[index], ")") == 0)
        {
           index++;
        }
else
{
error = 1;
printf("Error! ");
return;
}
}
else
```

```
{
error=1;
printf("Error! ");
return;
}

}

void main()
{

char inputstring[30];
char temp[30];

printf("Enter input string: ");
scanf("%[^\n]%*c",inputstring);

strcpy(temp, inputstring);

char *ptr = strtok(temp, " ");
while(ptr != NULL)
{
strcpy(productions[size++], ptr);
ptr = strtok(NULL, " ");
}


E();
if(size == index && error == 0)
{
printf("\n%s is accepted\n", inputstring);
}
else
{
printf("\n%s is rejected\n", inputstring);
}



}
```

Sample Output:

```
Enter input string: id + id * id
T() E() T() F() T'() T() E'() T() F() T'() F() T'() T() E'()
id + id * id is accepted

Enter input string: id + * id
T() E() T() F() T'() T() E'() T() F() Error! E'()
id + * id is rejected

Enter input string: ( id + id
T() E() T() F() T() E() T() F() T'() T() E'() T() F() T'() T() E'() Error! E'()
```

(id + id is rejected

Learning Outcomes:

This assignment helped me

1. To understand the concept of recursive descent parser.
2. To understand how a string is accpeted or rejected based on the rules.
3. To implement recursive descent parser.