Assignment No. 5

Write a program to implement a Recursive Descent Parser.

Recursive Descent Parser:
It is a kind of Top-Down Parser. A top-down parser builds the parse tree from the top to down, starting with the start non-terminal. A Predictive Parser is a special case of Recursive Descent Parser, where no Back Tracking is required.
By carefully writing a grammar means eliminating left recursion and left factoring from it, the resulting grammar will be a grammar that can be parsed by a recursive descent parser.

PROGRAM:

```
#include<stdio.h>

#include<conio.h>

#include<string.h>

char input[100];

int i,l;

void main()

{

clrscr();

printf("\nRecursive descent parsing for the following grammar\n"); printf("\nE->TE'\n
E'->+TE' |@\n     T->FT'\n T'->*FT'|@\n  F->(E)|ID\n");
printf("\nEnter the string to be checked:");
gets(input);

if(E())

{

if(input[i+1]=='\0')

printf("\nString is accepted");
```

```
else

printf("\nString is not accepted");

}

else

printf("\nString not accepted");

getch();

}

E()

{

if(T())

{

if(EP())

return(1);

else

return(0);

}

else

return(0);

}

EP()

{

if(input[i]=='+')

{
```

```
i++;
if(T())
{
if(EP())
return(1);
else
return(0);
}
else
return(0);
}
else
return(1);
}
T()
{
if(F())
{
if(TP())
return(1);
else
return(0);
```

```
}
else
return(0);
}
TP()
{
if(input[i]=='*')
{
i++;
if(F())
{
if(TP())
return(1);
else
return(0);
}
else
return(0);
}
else
return(1);
}
F()
```

```
{

if(input[i]=='(')

{

i++;

if(E())

{

if(input[i]==')')

{

i++;

return(1);

}

else

return(0);

}

else

return(0);

}

else if(input[i]>='a'&&input[i]<='z'||input[i]>='A'&&input[i]<='Z')

{

i++;

return(1);

}
else

return(0);

}
```

INPUT & OUTPUT:

```
Recursive descent parsing for the following grammar

E->TE'

E'->+TE' | @

T->FT'

T'->*FT | @

F->(E) | ID

Enter the string to be checked: (a+b)*c

String is accepted

Recursive descent parsing for the following grammar

E->TE'

E'->+TE' | @

T->FT'

T'->*FT | @

F->(E) | ID


Enter the string to be checked: a/c+d

String is not accepted
```

Conclusion:

Thus the program for implementation Recursive Descent Parser has been executed successfully.