Practical: 9

Aim: Write a program to implement Recursive Descendent Parsing (RDP) for the below given grammar.

$$E \rightarrow TE'$$

$$E' \rightarrow +TE'|\ ^$$

$$T \rightarrow FT'$$

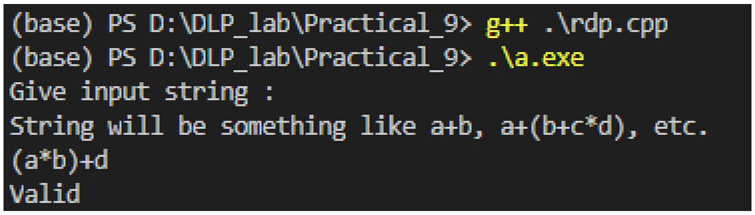
$$T' \rightarrow * \ FT'|\ ^$$

$$F \rightarrow (E)|\ id$$

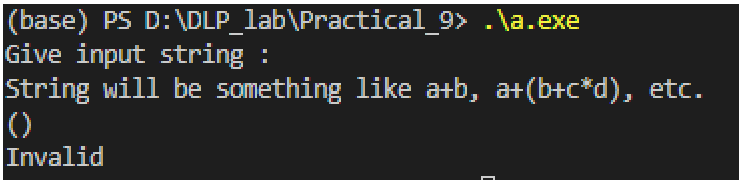
Program:

```
/*
    author: mr_bhishm
    created: 30-09-2020 19:49:18
    "Make it work, make it right, make it fast."
                                                – Kent Beck
*/

/*  :::RDP for Grammar given below:::
*
*   E->TE'
*   E'->+TE'|^
*   T->FT'
*   T'->*FT'|^
*   F->(E)|id
*
*/

#include<bits/stdc++.h>
using namespace std;
#define debug(x) cout<<#x<<" "<<x<<endl

string input;
char lookahead='$';
int len = 0;

void Match();
void E();
void EDS();
void T();
void TDS();
void F();

string RecursiveDescent(){
    Match();
    E();
    if(lookahead == '$'){
        return "Valid";
    }else{
        return "Invalid";
    }
}

void Match(){
    lookahead = input[len++];
}

void E(){
    T();
    EDS();
}

void EDS(){
    if(lookahead == '+') {
        Match();
        T();
        EDS();
    }else{
        return;
    }
}

void T(){
    F();
    TDS();
}

void TDS(){
    if(lookahead == '*') {
        Match();
        F();
        TDS();
    }else{
        return;
    }
}

void F(){
    if(lookahead == '('){
        Match();
        E();
        if(lookahead == ')'){
            Match();
        }
    }else{
        if(isalpha(lookahead)){
            Match();
        }else{
            lookahead = -1;
        }
    }
}



int main(){
    cout<<"Give input string : "<<endl;
    cout<<"String will be something like a+b, a+(b+c*d), etc."<<endl;
    cin>>input;
    input = input+"$";
    string ans = RecursiveDescent();
    cout<<ans<<endl;
}
```

Output:



```
(base) PS D:\DLP_lab\Practical_9> g++ .\rdp.cpp
(base) PS D:\DLP_lab\Practical_9> .\a.exe
Give input string :
String will be something like a+b, a+(b+c*d), etc.
(a*b)+d
Valid
```



```
(base) PS D:\DLP_lab\Practical_9> .\a.exe
Give input string :
String will be something like a+b, a+(b+c*d), etc.
()
Invalid
```

Conclusion: From this practical I have learnt about how to implement RDP for given grammar.