

Program:

```
#include<stdio.h>
#include<conio.h>
#include<stdlib.h>
char ip_sym[50];
int ip_ptr=0;
void e();
void e_prime();
void t();
void t_prime();
void f();
void advance();

void e(){
    printf("\nE--->TE");
    t();
    e_prime();
}
void e_prime(){
    if(ip_sym[ip_ptr]=='+'){
        printf("\nE'--->+TE");
        advance();
        t();
        e_prime();
    }
    else
        printf("\nE'--->e");
}
void t(){
    printf("\nT--->FT");
    f();
    t_prime();
}
void t_prime(){
    if(ip_sym[ip_ptr]=='*'){
        printf("\nT'--->+FT");
        advance();
        f();
        t_prime();
    }
    else
        printf("\nT'--->e");
}
void f(){
    if(ip_sym[ip_ptr]=='i' || ip_sym[ip_ptr]=='d') {
        printf("\nF--->id");
        advance();
    }
```

```

    }
    else{
        if(ip_sym[ip_ptr]=='('){
            advance();
            e();
            if(ip_sym[ip_ptr]==')'){
                advance();
                printf("\nF--->(E)");
            }
        }
        else{
            printf("\nSyntax Error...");
            getch();
            exit(1);
        }
    }
}
}
}
}

void advance(){
    ip_ptr++;
}

void main(){
    clrscr();
    printf("Recursive Descent Parser");
    printf("\nGRAMMER WITHOUT LEFT RECURSION");
    printf("\n\tE--->TE'\n\tE'--->+TE/e\n\tT---FT'\n\tT'--->*FT/e>\n\tF--->(E)/id");
    printf("\n\nEnter the input symbol :");
    gets(ip_sym);
    printf("\nSequence of production Rules");
    e();
    getch();
}

```

Sample Input & Output:

```

Recursive Descent Parser
GRAMMER WITHOUT LEFT RECURSION
    E--->TE'
    E'--->+TE/e
    T---FT'
    T'--->*FT/e>
    F--->(E)/id

Enter the input symbol :T

Sequence of production Rules
E--->TE'
T--->FT'
T'--->e
E'--->e

```