

```

print("Recursive Descent Parsing For following grammar\n")
print("E->TE'\nE' -> +TE' / @\nT->FT'\nT' -> *FT' / @\nF->(E)/i\n")
print("Enter the string want to be checked\n")
global s
s=list(input())
global i
i=0
def match(a):
    global s
    global i
    if(i>=len(s)):
        return False
    elif(s[i]==a):
        i+=1
        return True
    else:
        return False
def F():
    if(match("(")):
        if(E()):
            if(match(")")):
                return True
            else:
                return False
        else:
            return False
    elif(match("i")):
        return True
    else:
        return False
def Tx():
    if(match("*")):
        if(F()):
            if(Tx()):
                return True
            else:
                return False
        else:
            return False
    else:

```

```

        return True
def T():
    if(F()):
        if(Tx()):
            return True
        else:
            return False
    else:
        return False
def Ex():
    if(match("+")):
        if(T()):
            if(Ex()):
                return True
            else:
                return False
        else:
            return False
    else:
        return True
def E():
    if(T()):
        if(Ex()):
            return True
        else:
            return False
    else:
        return False
if(E()):
    if(i==len(s)):
        print("String is accepted")
    else:
        print("String is not accepted")

else:
    print("string is not accepted")

```

===== RESTART: C:/Users/student.RCOESRV/Desktop/Exp 4 spcc.py =====

Recursive Descent Parsing For following grammar

$E \rightarrow TE'$

$E' \rightarrow +TE' / \epsilon$

$T \rightarrow FT'$

$T' \rightarrow *FT' / \epsilon$

$F \rightarrow (E) / i$

Enter the string want to be checked

i+i+i

String is accepted