```
#CMPS 455 Assignment No. 7 Pt. 2
#Authors: Koppany Horvath
#Language: Python 3.6
#Task: Write a program to determine which of the following are accepted or rejected by the grammar: (i) a=(a+a)*b$ (ii) a=a*(b-a)$ (iii) a=(a+a)b$

def matches(string):
      string = string.replace("a=","q") #use q to represent a=
      parseTable = {
            "S":{"q":"qE", "a":None, "b":None, "+":None,  "-":None, "*":None,  "/":None,  "(":None,  ")":None, "$":None},
            "E":{"q":None, "a":"TQ", "b":"TQ", "+":None,  "-":None, "*":None,  "/":None,  "(":"TQ",  ")":None, "$":None},
            "Q":{"q":None, "a":None, "b":None, "+":"+TQ", "-":"-TQ", "*":None,  "/":None,  "(":None,  ")":"",   "$":""},
            "T":{"q":None, "a":"FR", "b":"FR", "+":None,  "-":None, "*":None,  "/":None,  "(":"FR",  ")":None, "$":None},
            "R":{"q":None, "a":None, "b":None, "+":"",    "-":"", "*":"*FR", "/":"/FR", "(":None,  ")":"",   "$":""},
            "F":{"q":None, "a":"a",  "b":"b",  "+":None,  "-":None, "*":None,  "/":None,  "(":"(E)", ")":None, "$":None}
      }
      stack = []
      curTerm = None
      curNonTerm = None
      done = False
      isGood = True

      stack.append("$")
      stack.append("S")

      while not done:
            curTerm = string[0] #read
            string = string[1:]

            while 1:
                  curNonTerm = stack.pop()
                  if curNonTerm in "qab+-*/()$": #if it's a term, match
                        print("Match:", curNonTerm.replace("q","a="), " - ", "Stack:", stack)
                        if curNonTerm == "$": done = True #if it's the end then exit
                        break

                  p = parseTable[curNonTerm][curTerm]
                  if p == "": continue #if it's lambda, pop again
                  elif p == None: #if it's none, break with error
                        done = True
                        isGood = False
                        break

                  for x in p[::-1]: stack.append(x) #push in reverse order
      return isGood

for s in ["a=(a+a)*b$","a=a*(b-a)$","a=(a+a)b$"]:
```

```
47          print("Working on string:", s)
48          isMatch = matches(s)
49          if isMatch: print("String matches grammar!")
50          else: print("Error: string does not match grammar!")
51          print()
52
53      """ Output:
54      Working on string: a=(a+a)*b$
55      Match: a=  -  Stack: ['$', 'E']
56      Match: (  -  Stack: ['$', 'Q', 'R', ')', 'E']
57      Match: a  -  Stack: ['$', 'Q', 'R', ')', 'Q', 'R']
58      Match: +  -  Stack: ['$', 'Q', 'R', ')', 'Q', 'T']
59      Match: a  -  Stack: ['$', 'Q', 'R', ')', 'Q', 'R']
60      Match: )  -  Stack: ['$', 'Q', 'R']
61      Match: *  -  Stack: ['$', 'Q', 'R', 'F']
62      Match: b  -  Stack: ['$', 'Q', 'R']
63      Match: $  -  Stack: []
64      String matches grammar!
65
66      Working on string: a=a*(b-a)$
67      Match: a=  -  Stack: ['$', 'E']
68      Match: a  -  Stack: ['$', 'Q', 'R']
69      Match: *  -  Stack: ['$', 'Q', 'R', 'F']
70      Match: (  -  Stack: ['$', 'Q', 'R', ')', 'E']
71      Match: b  -  Stack: ['$', 'Q', 'R', ')', 'Q', 'R']
72      Match: -  -  Stack: ['$', 'Q', 'R', ')', 'Q', 'T']
73      Match: a  -  Stack: ['$', 'Q', 'R', ')', 'Q', 'R']
74      Match: )  -  Stack: ['$', 'Q', 'R']
75      Match: $  -  Stack: []
76      String matches grammar!
77
78      Working on string: a=(a+a)b$
79      Match: a=  -  Stack: ['$', 'E']
80      Match: (  -  Stack: ['$', 'Q', 'R', ')', 'E']
81      Match: a  -  Stack: ['$', 'Q', 'R', ')', 'Q', 'R']
82      Match: +  -  Stack: ['$', 'Q', 'R', ')', 'Q', 'T']
83      Match: a  -  Stack: ['$', 'Q', 'R', ')', 'Q', 'R']
84      Match: )  -  Stack: ['$', 'Q', 'R']
85      Error: string does not match grammar!
86      """
```