string=input()

string+="$"

inputList=list(string)

#x=len(inputList)

stack=["0"]

table={"0id":"S5","0(":"S4","1+":"S6","1$":"Accept","2+":"R2","2\*":"S7","2)":"R2","2$":"R2","3+":"R4","3\*":"R4","3)":"R4","3$":"R4","4id":"S5","4(":"S4","5+":"R6","5\*":"R6","5)":"R6","5$":"R6","6id":"S5","6(":"S4","7id":"S5","7(":"S4","8+":"S6","8)":"S11","9+":"R1","9\*":"S7","9)":"R1","9$":"R1","@+":"R3","@\*":"R3","@)":"R3","@$":"R3","#+":"R5","#\*":"R5","#)":"R5","#$":"R5"}

reduce=[["id","F"],["(E)","F"],["F","T"],["T\*F","T"],["T","E"],["E+T","E"]]

goto={"0E":"1","0T":"2","0F":"3","4E":"8","4T":"2","4F":"3","6T":"9","6F":"3","7F":"@"}

i=0 #i inputList counter @=10 #=11

k=0 #k stack counter

flag=0 #is input id or not

flag1=0 #syntax error not occured

while stack[0]!="1" or stack[1]!="$":

if k!=0:

if inputList[i]!="i":

if ((stack[k-1]).isdigit==0) or (stack[k-1]=="+"or"(" or ")"or"\*"):

newTerm=stack[k]+inputList[i]

else:

newTerm=stack[k-1]+stack[k]+inputList[i]

if inputList[i]=="i":

if ((stack[k-1]).isdigit==0) or (stack[k-1]=="+"or"(" or ")"or"\*"):

newTerm=stack[k]+inputList[i]+inputList[i+1]

else:

newTerm=stack[k-1]+stack[k]+inputList[i]+inputList[i+1]

flag+=1

if k==0:

if inputList[0]=="i":

newTerm="0"+"id"

flag+=1

else:

newTerm="0"+inputList[0]

if newTerm=="1$":

print("VALID string entered. ACCEPTED!")

break

try:

newAction=table[newTerm]

except KeyError:

print("INVALID string entered. SYNTAX ERROR!")

break

newNumber=newAction[1:]

print(newTerm,newAction)

y=len(newNumber)

if "S" in newAction:

if flag==1:

stack.insert(k+1,inputList[i])

stack.insert(k+2,inputList[i+1])

i+=2

if y==2:

stack.insert(k+3,newNumber[0])

stack.insert(k+4,newNumber[1])

k+=4

else:

stack.insert(k+3,newNumber[0])

k+=3

if flag!=1:

stack.insert(k+1, inputList[i])

i+=1

if y==2:

stack.insert(k+2, newNumber[0])

stack.insert(k+3, newNumber[1])

k+=3

else:

stack.insert(k+2, newNumber[0])

k+=2

else:

if stack[k-1]=="d":

stack.pop()

stack.pop()

stack.pop()

stack.insert(k-2,"F")

gotoNumb=stack[k-3]

newGoto=goto.get(gotoNumb+"F")

lenght=len(newGoto)

if lenght==2:

stack.insert(k-1,newGoto[0])

stack.insert(k,newGoto[1])

else:

stack.insert(k-1,newGoto[0])

k-=1

elif stack[k-1]==")":

stack.pop()

stack.pop()

stack.pop()

stack.pop()

stack.pop()

stack.pop()

stack.insert(k-5,"F")

gotoNumb=stack[k-6]

newGoto=goto.get(gotoNumb+"F")

lenght=len(newGoto)

if lenght==2:

stack.insert(k-4,newGoto[0])

stack.insert(k-3, newGoto[1])

k-=3

if lenght==1:

stack.insert(k-4, newGoto[0])

k-=4

elif stack[k-1]=="F" and stack[k-3]!="\*":

stack.pop()

stack.pop()

stack.insert(k-1,"T")

gotoNumb=stack[k-2]

newGoto=goto.get(gotoNumb+"T")

lenght=len(newGoto)

if lenght==2:

stack.insert(k, newGoto[0])

stack.insert(k+1,newGoto[1])

k+=1

if lenght==1:

stack.insert(k, newGoto[0])

elif (stack[k-1]=="F" and stack[k-3]=="\*"):

stack.pop()

stack.pop()

stack.pop()

stack.pop()

stack.pop()

stack.pop()

stack.insert(k-5,"T")

gotoNumb=goto.get(gotoNumb+"T")

stack.insert(k-4, newGoto[0])

k-=4

elif stack[k-1]=="T" and stack[k-3]!="+":

stack.pop()

stack.pop()

stack.insert(k-1,"E")

gotoNumb=stack[k-2]

newGoto=goto.get(gotoNumb+"E")

lenght=len(newGoto)

if lenght==2:

stack.insert(k,newGoto[0])

stack.insert(k,newGoto[1])

k+=1

if lenght==1:

stack.insert(k, newGoto[0])

elif stack[k-1]=="T" and stack[k-3]=="+":

stack.pop()

stack.pop()

stack.pop()

stack.pop()

stack.pop()

stack.pop()

stack.insert(k-5,"E")

gotoNumb=stack[k-6]

newGoto=goto.get(gotoNumb+"E")

lenght=len(newGoto)

if lenght==2:

stack.insert(k-2,newGoto[0])

stack.insert(k-1,newGoto[1])

k-=3

if lenght==1:

stack.insert(k-2,newGoto[0])

k-=4

flag=0

#E->E+T

#E->T

#T->T\*F

#T->F

#F->(E)

#F->id