```
lines = []
data = []

def check(var, data):
    if var.isalpha():
        # we're checking for a variable
        for i in range(len(data)):
            if len(data[i]) == 2:
                if data[i][0] == var:
                    return (True,i)
        return (False,-1)

    else:
        for i in range(len(data)):
            if data[i] == var:
                return (True,i)
        return (False,-1)


with open('input_file.txt', 'r') as f:
    lines = f.readlines() # read all lines into a list of strings
print (lines)
# lines = ['a=5+3']
for statement in lines: # each statement is on a separate line
    token_list = statement.strip().split('=',1) # split a statement into a
list of tokens
    var = token_list[0]
    assert token_list[1] == '='
    exp = token_list[1].strip().split()


    if len(exp) == 3:
        #term binaryop term
        a1 = exp.split()[0]
        a2 = exp.split()[1]  #operator
        a3 = exp.split()[2]

        # check if a1 and a3 are in the data list or not
        # if they are
        # if they are not
```

```
        # check if a1 or a3 is a variable or integer or true/false
        # Integer or True/False -> insert into data list if not present,
if present do nothing
        # if its a variable if not then error
        # the operator is a2

        (cond1, ind1) = check(a1,data)
        if cond1:
            if a1.isalpha():
                v1 = data[data[ind1][1]]
            else:
                v1 = data[ind1]
        else:
            if a1.isalpha():
                # raise error
                print("ERROR!!!")
                exit(1)
            else:
                data.append(a1)

        (cond2, ind2) = check(a3,data)
        if cond2:
            if a3.isalpha():
                v2 = data[data[ind2][1]]
            else:
                v2 = data[ind2]
        else:
            if a3.isalpha():
                print("ERROR!!!")
                exit(1)
            else:
                data.append(a3)

        # perform the operation using a2
        # newdata = v1 opr(a2) v2

        (cond3, ind3) = check(var)
        if cond3:
            data[ind3] = (var, newdata)
        else:
```

```
            data.append((var, newdata))

    if 1 <= len(exp) <= 2:
        #unaryop term
        if len(exp) == 1:
            a1 = exp[0]
            # check if a1 is a variable or integer or true/false
            # Integer or True/False -> insert into data list if not
present
            # if its a variable then check if its in the data list or not
            # if not then error
            (cond1, ind1) = check(a1,data)
            if cond1:
                if a1.isalpha():
                    v1 = data[data[ind1][1]]
                else:
                    v1 = data[ind1]
            else:
                if a1.isalpha():
                    print("ERROR!!!")
                    exit(1)
                else:
                    data.append(a1)

            newdata = v1
            (cond3, ind3) = check(var)
            if cond3:
                data[ind3] = (var, newdata)
            else:
                data.append((var, newdata))


        if len(exp) == 2:
            a2 = exp.split()[0]
            a1 = exp.split()[1]
            # a1 is the unary operator '-' or 'not'
            # check if a2 is a variable or integer or true/false
            # Integer or True/False -> insert into data list if not
present
            # if its a variable then check if its in the data list or not
```

```
            # if not then error
            (cond1, ind1) = check(a1,data)
            if cond1:
                if a1.isalpha():
                    v1 = data[data[ind1][1]]
                else:
                    v1 = data[ind1]
            else:
                if a1.isalpha():
                    print("ERROR!!!")
                    exit(1)
                else:
                    data.append(a1)

            # newdata = opr(a2) a1
            (cond3, ind3) = check(var)
            if cond3:
                data[ind3] = (var, newdata)
            else:
                data.append((var, newdata))
    else:
        # error invalid expression
        print("Invalid expression")
        exit(1)
    # print ("Tokens: ", token_list)
    refvar = []
    for i in range(len(data)):
        ele = data[i]
        if len(ele) == 2:
            print(f'var: {ele[0]}, value: {data[ele[1]]}')
            refvar.append(ele[1])

    for i in range(len(data)):
        if i not in refvar:
            # garbage index
            # check if data[i] is an intger_constant
            print(data[i])
```