```
# DWAYNE FRASER
# PROBLEM 1


##############################################################################
##############################################################################
def ed_read(filename, from_index = 0, to = -1):
    """ Opens and Reads a file From: [Start Index, Stop Index)
    Returns that String """
    #Opens File, Stores Content in Variable
    filename = open(filename, "r")
    filecontent = filename.read()

    #Reads File Content
    try:
        if to > len(filecontent):
            raise IndexError("Your Parameter Exceeds the File Length")
        if(to == -1):
            string = filecontent[from_index:]
            return string

        else:
            string = filecontent[from_index:to]
    except:
        print("Index Out of Bounds")

    return string
##############################################################################
##############################################################################
def ed_find(filename, search_str):
    """ Finds a String and Returns a list index of its occurences """
    #Opens File, Stores Content in Variable
    filename = open(filename, "r")
    filecontent = filename.read()

    find_list = []
    index = 0

    #Finds File Content
    while True:
        index = filecontent.find(search_str, index)
        if index < 0:
            break
        find_list.append(index)
        index += 1

    return find_list
##############################################################################
##############################################################################
def ed_replace(filename, search_str, replace_with, occurrence=-1):
    """ Finds the ocuurences of a String and Replaces it with another string based on ocurrence value given from user
input
```

```
    Returns number of occurences replaced """
    #Opens File, Stores Content in Variable

    filecontent = ed_read(filename)

    find_list = ed_find(filename, search_str)
    total_occurrences = len(find_list)

    #Replaces File Content
    if(occurrence > total_occurrences):
        # Does not replace if occurrence > total_occurrences
        return 0

    if(occurrence == -1):
        # Replaces all ocurrences
        filecontent = filecontent.replace(search_str, replace_with, total_occurrences)
        return total_occurrences

    if(occurrence >= 0):
        # Replaces by occurence value
        filecontent = filecontent.replace(search_str, replace_with, occurrence)
        return total_occurrences

    return
################################################################################
################################################################################
def ed_append(filename,string):
    """appends string to the end of the file. If the file does not exist, a
    new file is created with the given file name. The function returns the number of characters
    written to the file."""
    #Opens File, Appends String to end of file
    filename = open(filename, "a")
    filename.write(string)

    return len(string)
################################################################################
################################################################################
def testif(b, testname, msgOK="", msgFailed=""):
    """Function used for testing.
    param b: boolean, normally a tested condition: true if test passed, false otherwise
    param testname: the test name
    param msgOK: string to be printed if param b==True  ( test condition true)
    param msgFailed: string to be printed if param b==False
    returns b
    """
    if b:
        print("Success: "+ testname + "; " + msgOK)
    else:
        print("Failed: "+ testname + "; " + msgFailed)
    return b
################################################################################
################################################################################
```

```
def test_ed_find(): # Testif for ed_find
    filename = "text.txt"
    search_str = "0"
    testif(ed_find(filename, search_str) == [4], "ed_find", "PASSED", "FAILED")

def test_ed_replace(): # Testif for ed_replace
    filename = "text.txt"
    search_str = "0"
    replace_with = "X"
    occurrence = -1
    testif(ed_replace(filename, search_str, replace_with, occurrence) == 1, "ed_replace", "PASSED", "FAILED")
################################################################################
################################################################################
def main():

    filename = "text.txt"

    print(ed_read(filename, 4, 7))       # Prints 0123
    print(ed_read(filename, 4, -1))      # Prints 01234

    test_ed_find()
    print(ed_find(filename, "0"))   # Prints 4
    print(ed_find(filename, "5"))   # Prints []

    test_ed_replace()
    print(ed_replace(filename, "01234", "EFGHI", -1))  # Prints 1
    print(ed_replace(filename, "01234", "EFGHI", 5))   # Prints 0

    filename = "newfile.txt"        # File Does Not Exist
    ed_append(filename, "Dwayne")    # Creates New File
    ed_append(filename, "Fraser")    # Appends to New File
    print(ed_read(filename))        # Prints File Contents

main()
```

```
In [1]: runfile('C:/Program Files (x86)/Work/Python/Python Dwayne Solutions/HW 4/p1_Fraser_Dwayne.py', wdir='C:/Program Files (x86)/Work/Python/Python
Dwayne Solutions/HW 4')
012
01234
Success: ed_find; PASSED
[4]
[]
Success: ed_replace; PASSED
1
0
DwayneFraser

In [2]:
```