Name:

Show work for all questions.

This homework is meant to help you review for exam 2. Since the exam will be written, I recommend you first attempt all problems *without* running or writing any code on the computer. As with the exam, avoid using python features or syntax not discussed in class, and do not use any built-in python functions which trivialize a problem.

Written HW 2

---

1. Using a while-loop, write a script which asks the user for an input value x and outputs all even numbers less than x, each on a different line.

assumed int >= 0

```
X = int(input("Enter x"))
c = 0
while(c<x):
    print(c)
    c+=2
```

2. Determine the output of the following code:

```
1 words = ['do', 'not', 'panic', 'breath']
2 i = 0
3 while len(words[i]) < 5:
4     i = i + 1
5 print(words[i])
```

0 1 2

it prints: panic

3. Suppose the variables `m` and `n` have already been defined and m < n. Using **two for loops**, write code to prints out the (m – n) × n multiplication table of all numbers between m and n and `n` columns.

For example, if `m`=3 and `n`=5, then the output of the code should be the 2 × 5 table below.

3	6	9	12	15
4	8	12	16	20

```
for i in range(m, n):
    for j in range(1, n+1):
        print(i*j, end=' ')
    print()
```

4. Determine the output of the following code:

```
1 for x in range(3):
2     for y in range(2):
3         print('yay loops', end=' ')
4         if y >= x:
5             print(y, end=' ')
6             print(x*y)
7 print('loops are fun')
```

Pay attention to newlines!

yay loops 0 0
yay loops 1 0
yay loops yay loops 1 1
yay loops yay loops loops are fun

5. Assume `n` is a variable that has already been defined, and it is equal to some odd positive integer greater than or equal to five. Using **only 2 loops**, no string multiplication, and only the following print statements,

```
print(' ', end='') # prints a single space
print('*', end='') # prints a single star
print() # prints a newline
```

write code to print out the following shape:

n stars newline

```
*****
*   *
*****
```

here, shown for `n=5`

```
*******
*     *
*     *
*     *
*******
```

and here, shown for `n=7`

Your code should work for any value of `n`, representing the number of rows/columns. The first, middle, and last rows have `n` stars; rows before the middle row have stars only in the first and last columns; rows after the middle row have stars only in the center column.

```
for i in range(n):
    for j in range(n):
        if(i==0 or i==n-1 or i==n//2):
            print('*', end='')
        elif(i < n//2 and (j==0 or j==n-1)):
            print('*', end='')
        elif(i > n//2 and j==n//2):
            print('*', end='')
        else:
            print(' ', end='')
    print()
```

6. Write code to read integer input from the user and store the result in a list called `num_list` only if the number is odd or ends in 2. The end of user input is signaled by a `X`; this should not be part of the list.

   For example, a run of the program may look like this:

```
Enter an integer: 2
Enter an integer: 8
Enter an integer: 1
Enter an integer: 23
Enter an integer: X
```

   where the numbers and `X` are input provided by the user. For this sample run, after your code, the result of

```
print(num_list)
```

   should be [2, 1, 23]

```
num_list = []
while (True):
    n = input("Enter an integer: ")
    if (n == 'X'):
        break
    n = int(n)
    if (n % 2 == 1 or n % 10 = 2):
        num_list.append(n)
print(num_list)
```

7. Write a program that opens a text file named `report.txt`, and prints out the longest line contained in that file.

```
X = open('report.txt', 'r')
max = 0
for i in X:
    if (len(i) > len(max)):
        max = i
X.close()
print(max)
```

8. Write code that receives a file name as input from the user. If the file exists, the code should open the file and print out the first word, then print out *every other word in the file* on a single line, separated by single spaces. For example, if the file's contents were

```
hi i am your written homework
use me to study for the exam
```

then, your code should print

```
hi am written use to for exam
```

If the file does not exist in the current directly, your code should display an error message to the user saying `the file does not exist` and keep asking for a file name until the user enters a valid file name.

```
while True:
    x = input("Give me a file name.")
```

```
while True:
    try:
        x = input("Give me a file name.")
        f = open(x, 'r')
        break
    except:
        print("file does not exist")
```

```
for i in f:
    s = i.split()
```

```
c = f.read()
c = c.split()
for i in range(0, len(c), 2):
    print(c[i], end = ' ')
f.close()
```

11. A nested list of integers is given as follows:

```
numlist = [[5, 1, 2, 4], [1, 3
```

9. Suppose you have a list called `mylist` already defined in Python. Write a code that outputs the alternating sum of the internal lists. For example, if `mylist` is defined below

```
1 mylist = [[2, 4], [3, 6, 9], [4, 8]] #content will change
```

then your code would output 0 since $(2+4)-(3+6+9)+(4+8)=0$. note that the 2d list is not necessarily square.

```
sum = 0
for i in mylist:
    for j in mylist[i]:
        if(i%2==0):
            sum += j
        else:
            sum -= j

print(sum)
```

10. What will print by the following code? Justify your answer.

```
1  def fn(my_num, my_list):
2      my_num += 1   # changes copy
3      my_list[0] += 1   # mutates ref works
4      return my_num*2
5      return my_num   # will never happen
6
7
8  a_number = 10
9  a_list = [4,8,2,3]
10 x = fn(a_number, a_list)
11 print(a_number)
12 print(a_list)
13 print(x)
```

(10+1)·2=22 → x = 22, a_list = [5,8,2,3]

4+1=5

10
[5,8,2,3]
22

11. A nested list of integers is given as follows:

```
numlist = [[5, 1, 2, 4], [1, 3, 1, 0, 6], [8, 1]]
```

Write a function `modify_list` so that after the function call `modify_list(numlist)`, each row consists of one entry equal to the average of the entries in that row.

For example, with the given `numlist`, after calling `modify_list(numlist)`, `numlist` should be changed to

```
[[3.0], [2.2], [4.5]]
```

Since the average of 5, 1, 2, 4 equals 3, the average of 1, 3, 1, 0, 6 equals 2.2, and the average of 8, 1 is 4.5.

Your code should work for any 2D list of integers named `numlist`.

```
def modify_list(numlist):
    for i in numlist:
        length = len(i)
        sum = 0
        for j in i:
            sum += j
        i.remove(i)
```

```
def modify_list(x):
    for i in range(len(x)):
        sum = 0
        length = len(x[i])
        for j in range(length):
            sum += x[i][j]
        x[i] = [sum/length]
```

13. Given are two lists of strin-
second is a list of
have

2. Write a function called `reverseList` that takes as input an even length list. After the function is applied to the list, the list should be *reversed*; the function should cause a side-effect on the input list. The function should not return anything.

For example, if `x=[1, 2, 3, 4, 5, 6]`, then after the function call `reverseList(x)` the list `x` should be `[6,5,4,3,2,1]`.

```
def reverseList(x):
    for i in range(len(x)//2):
        temp = x[i]
        x[i] = x[len(x)-1-i]
        x[len(x)-1-i] = temp
```

Or

`x[i], x[len(x)-1-i] = x[len(x)-1-i], x[i]`

13. Given are two lists of strings named `fruits` and `prices`. The first is a list of fruits for sale at a store; the second is a list of how much each fruit costs at my store (in the corresponding order). For example, we could have `fruits = ['Pear', 'Lime, 'Banana']` and `prices = [1.25, 0.85, 1.50]` (here, a pear costs $1.25, a lime costs $0.85, and a banana costs $1.50).

Write code that creates a dictionary called `fp`. This dictionary should contain the entries of fruits as keys, with the corresponding entries of prices as values. Your code should work with any two lists of strings named fruits and prices of the same length.

```
fp = {}
for i in range(len(fruits)):
    fp[fruits[i]] = prices[i]
```