

Homework 2

In this homework you will complete a couple of simple exercises in order to show your understanding with Python. If these exercises are challenging or new to you, you may want to reconsider taking the class and/or brush up on your Python skills. For the following exercises you are not allowed to use any Python packages (i.e. Numpy, Pandas, etc.).

NAME : Miran Tafazzul Hussain Junaidi

ID : 3034487132

Mandatory : Please print the output of each question below your code

Lists

1.1 Create an empty Python list called 'a' in the cell below.

In [1]:

```
#your code here  
a = []
```

1.2 Store all values between 1-100 (inclusive) with increments of 3 (i.e. 1, 4, 7...) in 'a'.

In [20]:

```
#your code here  
for i in range(1,100,3):  
    a.append(i)  
print(a)
```

```
[2, 8, 14, 20, 26, 32, 38, 44, 50, 56, 62, 68, 74, 80, 86, 92, 98,  
1, 4, 7, 10, 13, 16, 19, 22, 25, 28, 31, 34, 37, 40, 43, 46, 49, 52,  
55, 58, 61, 64, 67, 70, 73, 76, 79, 82, 85, 88, 91, 94, 97]
```

1.3 Create another list called 'a2' with numbers from 2-46 (inclusive) with increments of 0.5 (i.e. 2, 2.5, 3...).

In [21]:

```
#your code here
```

```
a2=[2]
```

```
i=0
```

```
while a2[i] < 46:
```

```
    a2.append(a2[i]+0.5)
```

```
    i+=1
```

```
print(a2)
```

```
[2, 2.5, 3.0, 3.5, 4.0, 4.5, 5.0, 5.5, 6.0, 6.5, 7.0, 7.5, 8.0, 8.5,
9.0, 9.5, 10.0, 10.5, 11.0, 11.5, 12.0, 12.5, 13.0, 13.5, 14.0, 14.
5, 15.0, 15.5, 16.0, 16.5, 17.0, 17.5, 18.0, 18.5, 19.0, 19.5, 20.0,
20.5, 21.0, 21.5, 22.0, 22.5, 23.0, 23.5, 24.0, 24.5, 25.0, 25.5, 2
6.0, 26.5, 27.0, 27.5, 28.0, 28.5, 29.0, 29.5, 30.0, 30.5, 31.0, 31.
5, 32.0, 32.5, 33.0, 33.5, 34.0, 34.5, 35.0, 35.5, 36.0, 36.5, 37.0,
37.5, 38.0, 38.5, 39.0, 39.5, 40.0, 40.5, 41.0, 41.5, 42.0, 42.5, 4
3.0, 43.5, 44.0, 44.5, 45.0, 45.5, 46.0]
```

1.4 Double every even integer element from list 'a'. Store the results back in 'a'.

In [22]:

```
#your code here
```

```
for i in range(len(a)):
```

```
    a[i] = a[i]*2
```

```
print(a)
```

```
[4, 16, 28, 40, 52, 64, 76, 88, 100, 112, 124, 136, 148, 160, 172, 1
84, 196, 2, 8, 14, 20, 26, 32, 38, 44, 50, 56, 62, 68, 74, 80, 86, 9
2, 98, 104, 110, 116, 122, 128, 134, 140, 146, 152, 158, 164, 170, 1
76, 182, 188, 194]
```

1.5 Add all numbers in 'a' except for the 2nd and 21st elements (the 2nd element here means the element at list index 1 and similarly for the 21st element).

In [5]:

```
#your code here
```

```
sum = 0
```

```
for i in range(len(a)):
```

```
    if i!= 1 and i!= 20:
```

```
        sum += a[i]
```

```
    else:
```

```
        pass
```

```
sum
```

Out[5]:

3104

1.6 Calculate the mean of 'a'.

In [6]:

```
#your code here
xi = 0
for i in a:
    xi += i
length = len(a)
mean = xi/length
mean
```

Out[6]:

98.0

1.7 Delete all elements greater than the mean value from list 'a'

In [7]:

```
#your code here
i=0
while i != (len(a)):
    if a[i]>mean:
        del a[i]
    else:
        i+=1
a
```

Out[7]:

[2, 8, 14, 20, 26, 32, 38, 44, 50, 56, 62, 68, 74, 80, 86, 92, 98]

Strings

2.1 Create an empty list called 'b'.

In [8]:

```
#your code here
b = []
```

2.2 Store the words in the sentence below as elements into the list 'b'.

In [23]:

```
#your code here
sentence = 'I am so excited about Data-X. It is important to be able to work with data.'
for words in sentence.split():
    b.append(words)
print(b)
```

```
['I', 'am', 'so', 'excited', 'about', 'Data-X.', 'It', 'is', 'important', 'to', 'be', 'able', 'to', 'work', 'with', 'data.']
```

2.3 Return the count of the occurrences of the lower-case letter 'e' in the list 'b'.

In [10]:

```
#your code here
count_of_e = 0
for i in b:
    for j in i:
        if j=='e':
            count_of_e +=1
count_of_e
```

Out[10]:

4

2.4 Replace every lower- or upper-case letter 'i' in the list b with a '1'.

In [24]:

```
#your code here
for i,j in enumerate(b):
    word = b[i].replace("I", "1")
    word = word.replace("i", '1')
    b[i] = word
print(b)
```

```
['1', 'am', 'so', 'exclted', 'about', 'Data-X.', 'lt', 'ls', 'lmpor-
ant', 'to', 'be', 'able', 'to', 'work', 'wlth', 'data.', 'Thls', '1
s', 'the', 'end', 'of', 'the', 'flrst', 'HW.', '1', 'am', 'so', 'exc
lted', 'about', 'Data-X.', 'lt', 'ls', 'lmporant', 'to', 'be', 'abl
e', 'to', 'work', 'wlth', 'data.']
```

2.5 Append the string "This is the end of the first HW." to the list 'b'.

In [25]:

```
#your code here
append_sentence = "This is the end of the first HW."
for i in append_sentence.split():
    b.append(i)
print(b)
```

```
['1', 'am', 'so', 'exclted', 'about', 'Data-X.', 'lt', 'ls', 'lmpor-
ant', 'to', 'be', 'able', 'to', 'work', 'wlth', 'data.', 'Thls', '1
s', 'the', 'end', 'of', 'the', 'flrst', 'HW.', '1', 'am', 'so', 'exc
lted', 'about', 'Data-X.', 'lt', 'ls', 'lmporant', 'to', 'be', 'abl
e', 'to', 'work', 'wlth', 'data.', 'This', 'is', 'the', 'end', 'of',
'the', 'first', 'HW.']
```

2.6 Print 'b' as ONE string backwards (starting with "WH tsrif...").

In [13]:

```
#your code here
string=''
for i in range(len(b)-1,-1,-1):
    for j in range (len(b[i])-1,-1,-1):
        string=string+b[i][j]
    string=string+' '
string
```

Out[13]:

```
'.WH tsrif eht fo dne eht si siHT .atad htlw krow ot elba eb ot tnat
ropml sl tl .X-ataD tuoba detlcxe os ma l '
```

Dictionaries

3.1 Put the following in a dictionary called 'codes':

Keys: 1001, 1002, 1003, 1004, 1005

Values: 'Alpha', 'Beta', 'Gamma', 'Delta', 'Tau'

then traverse the dictionary by its keys and change every value to be all lower case.

In [14]:

```
#your code here
codes = {1001: 'Alpha' , 1002 : 'Beta' , 1003 : 'Gamma' , 1004 : 'Delta' , 1005
: 'Tau'}

for key in codes:
    codes[key] = codes[key].lower()
codes
```

Out[14]:

```
{1001: 'alpha', 1002: 'beta', 1003: 'gamma', 1004: 'delta', 1005: 't
au'}
```

3.2 Delete 'alpha' from the dictionary.

In [15]:

```
#your code here
del codes[1001]
codes
```

Out[15]:

```
{1002: 'beta', 1003: 'gamma', 1004: 'delta', 1005: 'tau'}
```

Sets

4.1 Create a set called 'c' with the all the odd numbers less than 10.

In [16]:

```
#your code here
c=set([])
for i in range(1,10,2):
    c.add(i)
c
```

Out[16]:

```
{1, 3, 5, 7, 9}
```

4.2 Create another set called 'd' with elements 2, 5, 10, 30.

In [17]:

```
#your code here
d = {2,5,10,30}
```

4.3 Find the union between sets 'c' and 'd' and store this in a new set called 'e'.

In [18]:

```
#your code here
e = set([])
e= c.union(d)
e
```

Out[18]:

```
{1, 2, 3, 5, 7, 9, 10, 30}
```

4.4 Find the intersection between sets 'c' and 'd'.

In [19]:

```
#your code here
intersect = set([])
intersect = c.intersection(d)
intersect
```

Out[19]:

```
{5}
```

In []: