Part 1: Primer

1: Lists

1) Make a list with the spelled-out number strings ‘one’, ‘two’, ‘three’, ‘four’, and ‘five’ in that order and call it myList.

In [1]:
```
myl=["zero","one","two","three","four","five"]
print(myl)
```

```
['zero', 'one', 'two', 'three', 'four', 'five']
```

2) Remove ‘three’ from the list using positional indexing.

In [2]:
```
def deletingtheelement(lis,p):
    if p<len(lis):
        lis.pop(p)
l_o_n=["one","two","three","four","five"]
deletingtheelement(l_o_n,2)
print(l_o_n)
```

```
['one', 'two', 'four', 'five']
```

3) Check if ‘four’ is in the list.

In [3]:
```
numbers=["one","two","three","four","five"]
for k in numbers:
    if k=="four":
        print("four is in the list")
```

```
four is in the list
```

4) Append ‘six’ to the end of the list, then print the length of the list.

In [4]:
```
list=["one","two","three","four","five"]
list.append("six")
len(list)
```

Out[4]:
```
6
```

5) Print the contents of the list, but also next to each item print the length of the string (e.g. one is 3, four is 4) using a for loop.

In [5]:

```
list_of_numbers=["one","two","three","four"]
for k in list_of_numbers:
    print(k+" is",len(k))
```

```
one is 3
two is 3
three is 5
four is 4
```

6) Create a list only of the lengths of the strings and show your result. You can use the loop before to fill the list.

In [6]:

```
l_o_n=["one","two","three","four","five"]
list_length=[]
for j in l_o_n:
    list_length.append(len(j))
print(list_length)
```

```
[3, 3, 5, 4, 4]
```

2: Dictionaries

1) Make a dictionary with the keys be English words as below, and the values be the translation. You can use this language example (German) or choose your own. Note: you need to make sure all of these words are represented as strings, in quotes.

In [7]:

```
thisdict = {"apple": "Apfel","apples": "Äpfel","I": "Ich","and": "und","like": "
```

2) Use the dictionary to look up the translation for ‘apple’ and ‘like’.

In [8]:

```
print(thisdict["apple"])
print(thisdict["and"])
```

```
Apfel
und
```

3) Make a variable var with the string “I like apples and strawberries”.

In [9]:

```
var="I like apples and strawberries"
print(var)
```

```
I like apples and strawberries
```

4) Now create a list from var with each word a separate item (this is a string split operation).

In [10]:

```
vlist = var.split(" ")
print(vlist)
```

```
['I', 'like', 'apples', 'and', 'strawberries']
```

5) Iterate through the list you've created and replace any word in your dictionary with the translation.

In [11]:

```
for i in thisdict:
    for k in range(0,len(vlist)):
        if i==vlist[k]:
            vlist[k]=thisdict[i]
```

In [12]:

```
t=" "
t=t.join(vlist)
print(t)
```

```
Ich mag Äpfel und Erdbeeren
```

6) Now take your new list and turn it into a string with spaces between the words.

3.Arrays

1) Create an array of zeros of size 8 x 8 and print the data type of the array.

In [13]:

```
import numpy as np
a = np.zeros([8,8])
print(type(a))
print(type(a[0][0]))
```

```
<class 'numpy.ndarray'>
<class 'numpy.float64'>
```

2) Fill the array with the numbers 1 to 64 first by row, then by column. You may want to use a for loop inside a for loop to do this.

In [16]:

```
l =1
for m in range(0,len(a)):
    for n in range(0,len(a)):
        a[m][n]=int(l)
        l=l+1
a = a.astype(int)
print(a)
```

```
[[ 1  2  3  4  5  6  7  8]
 [ 9 10 11 12 13 14 15 16]
 [17 18 19 20 21 22 23 24]
 [25 26 27 28 29 30 31 32]
 [33 34 35 36 37 38 39 40]
 [41 42 43 44 45 46 47 48]
 [49 50 51 52 53 54 55 56]
 [57 58 59 60 61 62 63 64]]
```

In [19]:

```
l=int(1)
for m in range(0,len(a)):
    for n in range(0,len(a)):
        a[m][n]=round(l,0)
        l=l+1
a=a.astype(int)
print(a)
```

```
[[ 1  2  3  4  5  6  7  8]
 [ 9 10 11 12 13 14 15 16]
 [17 18 19 20 21 22 23 24]
 [25 26 27 28 29 30 31 32]
 [33 34 35 36 37 38 39 40]
 [41 42 43 44 45 46 47 48]
 [49 50 51 52 53 54 55 56]
 [57 58 59 60 61 62 63 64]]
```

3) Transpose the array.

In [22]:

```
print(a.transpose())
```

```
[[ 1  9 17 25 33 41 49 57]
 [ 2 10 18 26 34 42 50 58]
 [ 3 11 19 27 35 43 51 59]
 [ 4 12 20 28 36 44 52 60]
 [ 5 13 21 29 37 45 53 61]
 [ 6 14 22 30 38 46 54 62]
 [ 7 15 23 31 39 47 55 63]
 [ 8 16 24 32 40 48 56 64]]
```

4) Print only the top 4 rows and columns.

In [23]:

```
print(a[0:4,0:4])
```

```
[[ 1  2  3  4]
 [ 9 10 11 12]
 [17 18 19 20]
 [25 26 27 28]]
```

5) Make a 1D array out of your 2D array with the numbers 1 to 64 in order (note the column vs row issue, you may need transposes.)

In [24]:

```
a=a.transpose()
print(a.flatten())
```

```
[ 1  9 17 25 33 41 49 57  2 10 18 26 34 42 50 58  3 11 19 27 35 43 51 59
  4 12 20 28 36 44 52 60  5 13 21 29 37 45 53 61  6 14 22 30 38 46 54 62
  7 15 23 31 39 47 55 63  8 16 24 32 40 48 56 64]
```

6) Now take that 1D array you made from before and reshape it back to the original 2D array.

In [25]:

```
a=a.reshape(8,8)
print(a)
```

```
[[ 1  9 17 25 33 41 49 57]
 [ 2 10 18 26 34 42 50 58]
 [ 3 11 19 27 35 43 51 59]
 [ 4 12 20 28 36 44 52 60]
 [ 5 13 21 29 37 45 53 61]
 [ 6 14 22 30 38 46 54 62]
 [ 7 15 23 31 39 47 55 63]
 [ 8 16 24 32 40 48 56 64]]
```

Part 2: Applications

1.Word Counts

First convert the string to a list with each word a separate item in the list. Hint: use a string split function for your language, and make sure it separates by “ ”.

In [27]:

```python
i=input()
lst=i.split(" ")
u=set(lst)
count=0
for num in u:
    count=0
    for nums in lst:
        if num==nums:
            count=count+1
    print(num,count)
```

```
how much wood would a woodchuck chuck if a woodchuck could chuck wood
could 1
a 2
woodchuck 2
chuck 2
if 1
how 1
wood 2
much 1
would 1
```

Adding an Array Border

As we will see later in the class, arrays can be used to represent images. In particular, black and white images can be represented by a 2D array, with one number (usually 0) representing black and another number (usually 1, but sometimes something else) representing white, and gray being everything in between. Here we just want to test your ability to handle a 2D array and add a border (filled with 0's) around it. In code this means creating an array of 0’s of width+2 by height+2 size and copying the original array into the middle appropriately (Note: do not use np.pad() function).

In [28]:

```
import numpy as np
def a_b(a):
    a1=np.zeros((a.shape[0]+2,a.shape[1]+2))
    for i in range(a.shape[0]):
        for j in range(a.shape[1]):
            a1[i+1,j+1]=a[i,j]
            print("original Array")
            a=a.astype(int)
            print(a)
            print()
            print("New array")
            a1=a1.astype(int)
            print(a1)
a_b(np.ones((5,4)))
```

```
original Array
[[1 1 1 1]
 [1 1 1 1]
 [1 1 1 1]
 [1 1 1 1]
 [1 1 1 1]]

New array
[[0 0 0 0 0 0]
 [0 1 0 0 0 0]
 [0 0 0 0 0 0]
 [0 0 0 0 0 0]
 [0 0 0 0 0 0]
 [0 0 0 0 0 0]
 [0 0 0 0 0 0]]
original Array
[[1 1 1 1]
 [1 1 1 1]
 [1 1 1 1]
```

In [ ]: