

```
#!/usr/bin/env python
# coding: utf-8
```

```
# In[1]:
```

```
#Question 1 - Binary Search for Values Equal to their Indexes
```

```
lst = [-2,0,2,3,6,7,9] #sorted list
key = 3 # we choose an integer of our choice

start = 0
end = len(lst) - 1 #minus one in order to get the index
                        #number of the last number in the list
mid = 0

while start <= end:
    mid = (start + end) // 2 #define midpoint of list
    # print(mid)
    if lst[mid] > key: # this is where we check if the guess is bigger than the key
        end = mid
    elif lst[mid] < key: # this is where we check if the guess is smaller than the key
        start = mid
    elif lst[mid] == key: # this is where we check if the guess is the same as the key
        print (f"You searched for {lst[mid]} and we found it in the index number {mid} of the list")
        break
else:
    print ("Not Found!")
```

```
# In[12]:
```

```
#Question 2 - Comparing Numeric Strings
```

```
init_a = '295'
init_b = '500'

a = init_a
b = init_b

# check strings validity
if a is None or b is None:
    print('One of the numbers is None. Please verify your numbers and run      again')
# check strings validity
elif a == '' or b == '':
    print('One of the numbers is empty. Please verify your numbers and run      again')
else:

    # fill with leading zeroes to make strings have equal length
    if len(a) < len(b):
        a = '0' * (len(b) - len(a)) + a
    elif len(b) < len(a):
        b = '0' * (len(a) - len(b)) + b

    print(f'We have a match up of {a} vs {b}!')
    # compare digits of strings one by one
    for i in range(len(a)):
        if a[i] < b[i]:
            print(init_b, 'is greater than', init_a, '!')
            break
        elif b[i] < a[i]:
            print(init_a, 'is greater than', init_b, '!')
            break
        if b[i] == a[i]:
            print(init_a, 'is equal to', init_b, '!')
```

```
# In[3]:
```

```
#Question 3 - Key Value Pairs
```

```
tval = 10
med = tval//2
slist = [3,4,1,2,9]
dlist = {}
dlist = dlist.fromkeys(slist)

for i in slist:
    if tval - i in slist or i == med and slist.count(med) > 1:
        dlist[i] = 1
    elif not tval-i in slist or i == med and slist.count(med) == 1:
        dlist[i] = 1
print('\n')
print("Key | Value")
```

```

print('\n'.join("{} | {}".format(k, v) for k, v in dlist.items()))
print('\n')
for i in dlist:
    if i and tval - i in slist:
        print(f'Matching pair: {i} & {tval - i}')
print('\n')

```

```

# In[1]:

```

```

#Question 4

```

```

import json
from collections import defaultdict
from difflib import get_close_matches

```

```

# p_name = input("Enter product name: ")
p_name = 'Met'

```

```

data = []
json_file=open ('products.json', encoding="utf8")
json_str=json_file.read()
json_data=json.loads(json_str)
p = p_name.lower()

```

```

brands = {}

```

```

for item in range(len(json_data)):
    if json_data[item]['name'] is None:
#         print(json_data[item])
        continue
    arr = json_data[item]['name'].split('-', maxsplit=1)
    # trim your strings to clean whitespaces
    arr = [a.strip() for a in arr]
#     print(arr)
    key = arr[0]
    value = arr[-1]
    # key exist, append to list
    if key in brands:
        brands[key].append(value)
    else:
        # create the first value for key
        brands[key] = [value]

```

```

while True:
    command = input("Enter product name or 's' to stop: ")
    if command == 's':
        break
    p_name = command
    close_matches = get_close_matches(p_name, brands.keys())
    if p_name in close_matches:
        print('Exact match.')
        break
    if len(close_matches) == 0:
        print('Sorry. No matched names. Please enter a new one.')
        continue
    print('Those are the matched names. Did you find your target?')
    for key in close_matches:
        print(key)

```

```

# In[ ]:

```