In [1]:

```
data = ['Ajay','Aishwarya','Divyan','Deb']
```

In [2]:

```
def input_string():
    x = True
    while x == True:
        search = input("\n\nEnter the name to search : ")
        x = search.isdigit()

        if(x == True):
            print("Input shouldn't be a digit !")

    return search
```

In [3]:

```
def input_digit():
    x = False
    while x == False:
        search = input("\n\nEnter the no. to search : ")
        x = search.isdigit()

        if(x == False):
            print("Input should be a digit !")

    return search
```

Sequential Search

In [4]:

```
def Sequential_Search(dlist, item):
    pos = 0
    found = False

    while pos < len(dlist) and not found:
        if dlist[pos] == item:
            found = True
        else:
            pos = pos + 1
    return found,pos


search = input_string()
a,b = Sequential_Search(data,search)

if(a == True):
    print("\n\t{} is {} in position {}\n".format(search,"found",str(b)))
else:
    print("\n\t{} is {}\n".format(search,"not found"))
```

```
Enter the name to search : 121
Input shouldn't be a digit !


Enter the name to search : 3432
Input shouldn't be a digit !


Enter the name to search : Deb

        Deb is found in position 3
```

Ternary Search

In [6]:

```
def ternarySearch(l, r, key, ar):
    while r >= l:

        # Find mid1 and mid2
        mid1 = l + (r-l) // 3
        mid2 = r - (r-l) // 3

        # Check if key is at any mid
        if key == ar[mid1]:
            return mid1
        if key == mid2:
            return mid2

        # Since key is not present at mid,
        # Check in which region it is present
        # Then repeat the search operation in that region
        if key < ar[mid1]:
            # key lies between l and mid1
            r = mid1 - 1
        elif key > ar[mid2]:
            # key lies between mid2 and r
            l = mid2 + 1
        else:
            # key lies between mid1 and mid2
            l = mid1 + 1
            r = mid2 - 1

    # key not found
    return -1

ar = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
l = 0
r = 9
key = int(input_digit())
p = ternarySearch(l, r, key, ar)
if(p!=-1):
    print("Index of", key, "is", p)
else :
    print("{} not found\n\n".format(key))
```

```
Enter the no. to search : sdkaj
Input should be a digit !


Enter the no. to search : 12as
Input should be a digit !


Enter the no. to search : 12
12 not found
```

Installing Covid

Python Package to get information regarding the novel corona virus provided by Johns Hopkins university

In [7]:

```
! pip install covid
```

```
Requirement already satisfied: covid in c:\users\royas\anaconda3\lib\site-pa
ckages (2.2.10)
Requirement already satisfied: requests in c:\users\royas\anaconda3\lib\site
-packages (from covid) (2.22.0)
Requirement already satisfied: beautifulsoup4 in c:\users\royas\anaconda3\li
b\site-packages (from covid) (4.8.2)
Requirement already satisfied: pydantic in c:\users\royas\anaconda3\lib\site
-packages (from covid) (1.5.1)
Requirement already satisfied: urllib3!=1.25.0,!=1.25.1,<1.26,>=1.21.1 in
c:\users\royas\anaconda3\lib\site-packages (from requests->covid) (1.25.8)
Requirement already satisfied: certifi>=2017.4.17 in c:\users\royas\anaconda
3\lib\site-packages (from requests->covid) (2019.11.28)
Requirement already satisfied: chardet<3.1.0,>=3.0.2 in c:\users\royas\anaco
nda3\lib\site-packages (from requests->covid) (3.0.4)
Requirement already satisfied: idna<2.9,>=2.5 in c:\users\royas\anaconda3\li
b\site-packages (from requests->covid) (2.8)
Requirement already satisfied: soupsieve>=1.2 in c:\users\royas\anaconda3\li
b\site-packages (from beautifulsoup4->covid) (1.9.5)
```

Importing the COVID Data

In [8]:

```
from covid import Covid
```

In [9]:

```
covid = Covid()
```

In [10]:

```
countries = covid.list_countries()
print("\n\n\tNo. OF Countries = {}\n".format(len(countries)))
# print(countries)
```

```


        No. OF Countries = 188

```

Storing countries in the list

In [11]:

```
country_list =[]
for i in countries:
    country_list.append(i['name'])
print(country_list)
```

```
['US', 'Brazil', 'Russia', 'United Kingdom', 'Spain', 'Italy', 'France', 'Ge
rmany', 'India', 'Turkey', 'Iran', 'Peru', 'Canada', 'Chile', 'China', 'Mexi
co', 'Saudi Arabia', 'Pakistan', 'Belgium', 'Qatar', 'Netherlands', 'Banglad
esh', 'Belarus', 'Ecuador', 'Sweden', 'Singapore', 'United Arab Emirates',
'Portugal', 'Switzerland', 'South Africa', 'Colombia', 'Indonesia', 'Irelan
d', 'Kuwait', 'Poland', 'Ukraine', 'Egypt', 'Romania', 'Israel', 'Japan', 'A
ustria', 'Dominican Republic', 'Philippines', 'Argentina', 'Afghanistan', 'P
anama', 'Denmark', 'Korea, South', 'Serbia', 'Bahrain', 'Kazakhstan', 'Oma
n', 'Czechia', 'Algeria', 'Nigeria', 'Armenia', 'Norway', 'Bolivia', 'Malays
ia', 'Moldova', 'Morocco', 'Ghana', 'Australia', 'Finland', 'Iraq', 'Cameroo
n', 'Azerbaijan', 'Honduras', 'Guatemala', 'Sudan', 'Luxembourg', 'Hungary',
'Tajikistan', 'Guinea', 'Uzbekistan', 'Senegal', 'Thailand', 'Djibouti', 'Gr
eece', 'Congo (Kinshasa)', "Cote d'Ivoire", 'Bosnia and Herzegovina', 'Bulga
ria', 'Gabon', 'El Salvador', 'Croatia', 'North Macedonia', 'Cuba', 'Estoni
a', 'Somalia', 'Iceland', 'Kyrgyzstan', 'Lithuania', 'Kenya', 'Sri Lanka',
'Slovakia', 'Maldives', 'New Zealand', 'Slovenia', 'Haiti', 'Venezuela', 'Ne
pal', 'Guinea-Bissau', 'Mali', 'Lebanon', 'Albania', 'Tunisia', 'Latvia', 'Z
ambia', 'Kosovo', 'Equatorial Guinea', 'Costa Rica', 'South Sudan', 'Ethiopi
a', 'Niger', 'Cyprus', 'Paraguay', 'Burkina Faso', 'Sierra Leone', 'Urugua
y', 'Andorra', 'Nicaragua', 'Central African Republic', 'Georgia', 'Jordan',
'Chad', 'Diamond Princess', 'Madagascar', 'San Marino', 'Malta', 'Congo (Bra
zzaville)', 'Jamaica', 'Tanzania', 'Sao Tome and Principe', 'West Bank and G
aza', 'Taiwan*', 'Togo', 'Cabo Verde', 'Rwanda', 'Mauritania', 'Mauritius',
'Vietnam', 'Montenegro', 'Uganda', 'Eswatini', 'Yemen', 'Liberia', 'Mozambiq
ue', 'Benin', 'Burma', 'Malawi', 'Mongolia', 'Guyana', 'Zimbabwe', 'Brunei',
'Cambodia', 'Syria', 'Trinidad and Tobago', 'Libya', 'Bahamas', 'Monaco', 'B
arbados', 'Comoros', 'Liechtenstein', 'Angola', 'Burundi', 'Eritrea', 'Botsw
ana', 'Bhutan', 'Antigua and Barbuda', 'Gambia', 'Saint Vincent and the Gren
adines', 'Timor-Leste', 'Grenada', 'Namibia', 'Laos', 'Belize', 'Fiji', 'Sai
nt Lucia', 'Dominica', 'Saint Kitts and Nevis', 'Holy See', 'Suriname', 'Sey
chelles', 'MS Zaandam', 'Western Sahara', 'Papua New Guinea', 'Lesotho']
```

Storing active cases in a list

In [12]:

```
active_list = []
for i in covid.get_data():
    active_list.append(i['active'])
print(active_list)
```

```
[1249471, 233880, 223992, 231422, 60411, 47986, 90390, 9958, 91239, 32149, 2
4060, 78238, 36033, 49938, 101, 16315, 25191, 40406, 32949, 32267, 40200, 33
247, 23150, 16733, 27155, 15543, 15589, 11652, 609, 12456, 17864, 17204, 111
3, 15229, 11229, 13198, 14589, 4913, 1876, 1617, 640, 6631, 11069, 9577, 121
54, 4432, 787, 770, 4621, 4846, 4884, 7384, 2360, 3090, 6064, 5259, 448, 735
6, 1382, 3320, 2272, 5161, 488, 963, 2307, 3265, 1681, 4037, 3703, 3402, 95,
1300, 1842, 1581, 731, 1650, 74, 1653, 1357, 2364, 1283, 529, 1333, 1749, 12
21, 83, 487, 167, 182, 1446, 3, 558, 378, 1139, 766, 154, 1311, 1, 8, 1386,
1012, 1000, 1145, 470, 441, 215, 77, 295, 271, 217, 866, 344, 978, 763, 88,
140, 487, 75, 406, 135, 31, 354, 731, 158, 222, 248, 48, 529, 306, 94, 391,
276, 305, 378, 75, 15, 212, 231, 104, 312, 2, 49, 0, 248, 109, 210, 98, 149,
85, 74, 157, 136, 72, 117, 1, 2, 75, 0, 59, 43, 4, 9, 61, 26, 52, 21, 0, 14,
25, 3, 5, 11, 0, 5, 9, 3, 0, 3, 0, 0, 0, 10, 2, 0, 7, 2, 0, 1]
```

Sorting the countries list in ascending order based on active cases

In [13]:

```
n = len(active_list)
for i in range(n-1):
    for j in range(0, n-i-1):
        if active_list[j] > active_list[j+1]:
            active_list[j], active_list[j+1] = active_list[j+1], active_list[j]
            country_list[j], country_list[j+1] = country_list[j+1], country_list[j]

print(country_list)
```

```
['Montenegro', 'Trinidad and Tobago', 'Eritrea', 'Timor-Leste', 'Belize', 'S
aint Lucia', 'Dominica', 'Saint Kitts and Nevis', 'Seychelles', 'Papua New G
uinea', 'New Zealand', 'Brunei', 'Lesotho', 'Mauritius', 'Cambodia', 'Surina
me', 'Western Sahara', 'Iceland', 'Antigua and Barbuda', 'Laos', 'Fiji', 'Mo
naco', 'Gambia', 'Grenada', 'MS Zaandam', 'Slovenia', 'Barbados', 'Namibia',
'Holy See', 'Saint Vincent and the Grenadines', 'Botswana', 'Taiwan*', 'Buru
ndi', 'Bhutan', 'Liechtenstein', 'Andorra', 'Bahamas', 'Diamond Princess',
'Vietnam', 'Angola', 'Libya', 'Comoros', 'Guyana', 'Thailand', 'Burma', 'Bur
kina Faso', 'West Bank and Gaza', 'Syria', 'Tunisia', 'Croatia', 'Benin', 'N
iger', 'Malta', 'Luxembourg', 'Liberia', 'China', 'Rwanda', 'Eswatini', 'Zim
babwe', 'Uruguay', 'Mongolia', 'Cyprus', 'Mozambique', 'Slovakia', 'Malawi',
'Georgia', 'Cuba', 'Estonia', 'Yemen', 'Togo', 'Albania', 'Kosovo', 'Jorda
n', 'Cabo Verde', 'Chad', 'Uganda', 'Zambia', 'Jamaica', 'Latvia', 'Tanzani
a', 'San Marino', 'Mauritania', 'Costa Rica', 'Nicaragua', 'Lithuania', 'Sao
Tome and Principe', 'Congo (Brazzaville)', 'Sierra Leone', 'Lebanon', 'Norwa
y', 'Mali', 'North Macedonia', 'Paraguay', 'Australia', 'Bosnia and Herzegov
ina', 'Madagascar', 'Kyrgyzstan', 'Switzerland', 'Austria', 'Uzbekistan', 'C
entral African Republic', 'Ethiopia', 'Sri Lanka', 'Korea, South', 'Denmar
k', 'Equatorial Guinea', 'Finland', 'South Sudan', 'Nepal', 'Venezuela', 'Ir
eland', 'Kenya', 'Guinea-Bissau', 'El Salvador', "Cote d'Ivoire", 'Hungary',
'Maldives', 'Bulgaria', 'Greece', 'Malaysia', 'Haiti', 'Somalia', 'Guinea',
'Japan', 'Senegal', 'Djibouti', 'Azerbaijan', 'Gabon', 'Tajikistan', 'Israe
l', 'Morocco', 'Iraq', 'Czechia', 'Congo (Kinshasa)', 'Algeria', 'Cameroon',
'Moldova', 'Sudan', 'Guatemala', 'Honduras', 'Panama', 'Serbia', 'Bahrain',
'Kazakhstan', 'Romania', 'Ghana', 'Armenia', 'Nigeria', 'Dominican Republi
c', 'Bolivia', 'Oman', 'Argentina', 'Germany', 'Philippines', 'Poland', 'Por
tugal', 'Afghanistan', 'South Africa', 'Ukraine', 'Egypt', 'Kuwait', 'Singap
ore', 'United Arab Emirates', 'Mexico', 'Ecuador', 'Indonesia', 'Colombia',
'Belarus', 'Iran', 'Saudi Arabia', 'Sweden', 'Turkey', 'Qatar', 'Belgium',
'Bangladesh', 'Canada', 'Netherlands', 'Pakistan', 'Italy', 'Chile', 'Spai
n', 'Peru', 'France', 'India', 'Russia', 'United Kingdom', 'Brazil', 'US']
```

Reversing the country list

In [14]:

```
rev = country_list[::-1]
print(rev)
```

```
['US', 'Brazil', 'United Kingdom', 'Russia', 'India', 'France', 'Peru', 'Spa
in', 'Chile', 'Italy', 'Pakistan', 'Netherlands', 'Canada', 'Bangladesh', 'B
elgium', 'Qatar', 'Turkey', 'Sweden', 'Saudi Arabia', 'Iran', 'Belarus', 'Co
lombia', 'Indonesia', 'Ecuador', 'Mexico', 'United Arab Emirates', 'Singapor
e', 'Kuwait', 'Egypt', 'Ukraine', 'South Africa', 'Afghanistan', 'Portugal',
'Poland', 'Philippines', 'Germany', 'Argentina', 'Oman', 'Bolivia', 'Dominic
an Republic', 'Nigeria', 'Armenia', 'Ghana', 'Romania', 'Kazakhstan', 'Bahra
in', 'Serbia', 'Panama', 'Honduras', 'Guatemala', 'Sudan', 'Moldova', 'Camer
oon', 'Algeria', 'Congo (Kinshasa)', 'Czechia', 'Iraq', 'Morocco', 'Israel',
'Tajikistan', 'Gabon', 'Azerbaijan', 'Djibouti', 'Senegal', 'Japan', 'Guine
a', 'Somalia', 'Haiti', 'Malaysia', 'Greece', 'Bulgaria', 'Maldives', 'Hunga
ry', "Cote d'Ivoire", 'El Salvador', 'Guinea-Bissau', 'Kenya', 'Ireland', 'V
enezuela', 'Nepal', 'South Sudan', 'Finland', 'Equatorial Guinea', 'Denmar
k', 'Korea, South', 'Sri Lanka', 'Ethiopia', 'Central African Republic', 'Uz
bekistan', 'Austria', 'Switzerland', 'Kyrgyzstan', 'Madagascar', 'Bosnia and
Herzegovina', 'Australia', 'Paraguay', 'North Macedonia', 'Mali', 'Norway',
'Lebanon', 'Sierra Leone', 'Congo (Brazzaville)', 'Sao Tome and Principe',
'Lithuania', 'Nicaragua', 'Costa Rica', 'Mauritania', 'San Marino', 'Tanzani
a', 'Latvia', 'Jamaica', 'Zambia', 'Uganda', 'Chad', 'Cabo Verde', 'Jordan',
'Kosovo', 'Albania', 'Togo', 'Yemen', 'Estonia', 'Cuba', 'Georgia', 'Malaw
i', 'Slovakia', 'Mozambique', 'Cyprus', 'Mongolia', 'Uruguay', 'Zimbabwe',
'Eswatini', 'Rwanda', 'China', 'Liberia', 'Luxembourg', 'Malta', 'Niger', 'B
enin', 'Croatia', 'Tunisia', 'Syria', 'West Bank and Gaza', 'Burkina Faso',
'Burma', 'Thailand', 'Guyana', 'Comoros', 'Libya', 'Angola', 'Vietnam', 'Dia
mond Princess', 'Bahamas', 'Andorra', 'Liechtenstein', 'Bhutan', 'Burundi',
'Taiwan*', 'Botswana', 'Saint Vincent and the Grenadines', 'Holy See', 'Nami
bia', 'Barbados', 'Slovenia', 'MS Zaandam', 'Grenada', 'Gambia', 'Monaco',
'Fiji', 'Laos', 'Antigua and Barbuda', 'Iceland', 'Western Sahara', 'Surinam
e', 'Cambodia', 'Mauritius', 'Lesotho', 'Brunei', 'New Zealand', 'Papua New
Guinea', 'Seychelles', 'Saint Kitts and Nevis', 'Dominica', 'Saint Lucia',
'Belize', 'Timor-Leste', 'Eritrea', 'Trinidad and Tobago', 'Montenegro']
```

Copying the country list

In [15]:

```
copy = rev
print(copy)
```

```
['US', 'Brazil', 'United Kingdom', 'Russia', 'India', 'France', 'Peru', 'Spa
in', 'Chile', 'Italy', 'Pakistan', 'Netherlands', 'Canada', 'Bangladesh', 'B
elgium', 'Qatar', 'Turkey', 'Sweden', 'Saudi Arabia', 'Iran', 'Belarus', 'Co
lombia', 'Indonesia', 'Ecuador', 'Mexico', 'United Arab Emirates', 'Singapor
e', 'Kuwait', 'Egypt', 'Ukraine', 'South Africa', 'Afghanistan', 'Portugal',
'Poland', 'Philippines', 'Germany', 'Argentina', 'Oman', 'Bolivia', 'Dominic
an Republic', 'Nigeria', 'Armenia', 'Ghana', 'Romania', 'Kazakhstan', 'Bahra
in', 'Serbia', 'Panama', 'Honduras', 'Guatemala', 'Sudan', 'Moldova', 'Camer
oon', 'Algeria', 'Congo (Kinshasa)', 'Czechia', 'Iraq', 'Morocco', 'Israel',
'Tajikistan', 'Gabon', 'Azerbaijan', 'Djibouti', 'Senegal', 'Japan', 'Guine
a', 'Somalia', 'Haiti', 'Malaysia', 'Greece', 'Bulgaria', 'Maldives', 'Hunga
ry', "Cote d'Ivoire", 'El Salvador', 'Guinea-Bissau', 'Kenya', 'Ireland', 'V
enezuela', 'Nepal', 'South Sudan', 'Finland', 'Equatorial Guinea', 'Denmar
k', 'Korea, South', 'Sri Lanka', 'Ethiopia', 'Central African Republic', 'Uz
bekistan', 'Austria', 'Switzerland', 'Kyrgyzstan', 'Madagascar', 'Bosnia and
Herzegovina', 'Australia', 'Paraguay', 'North Macedonia', 'Mali', 'Norway',
'Lebanon', 'Sierra Leone', 'Congo (Brazzaville)', 'Sao Tome and Principe',
'Lithuania', 'Nicaragua', 'Costa Rica', 'Mauritania', 'San Marino', 'Tanzani
a', 'Latvia', 'Jamaica', 'Zambia', 'Uganda', 'Chad', 'Cabo Verde', 'Jordan',
'Kosovo', 'Albania', 'Togo', 'Yemen', 'Estonia', 'Cuba', 'Georgia', 'Malaw
i', 'Slovakia', 'Mozambique', 'Cyprus', 'Mongolia', 'Uruguay', 'Zimbabwe',
'Eswatini', 'Rwanda', 'China', 'Liberia', 'Luxembourg', 'Malta', 'Niger', 'B
enin', 'Croatia', 'Tunisia', 'Syria', 'West Bank and Gaza', 'Burkina Faso',
'Burma', 'Thailand', 'Guyana', 'Comoros', 'Libya', 'Angola', 'Vietnam', 'Dia
mond Princess', 'Bahamas', 'Andorra', 'Liechtenstein', 'Bhutan', 'Burundi',
'Taiwan*', 'Botswana', 'Saint Vincent and the Grenadines', 'Holy See', 'Nami
bia', 'Barbados', 'Slovenia', 'MS Zaandam', 'Grenada', 'Gambia', 'Monaco',
'Fiji', 'Laos', 'Antigua and Barbuda', 'Iceland', 'Western Sahara', 'Surinam
e', 'Cambodia', 'Mauritius', 'Lesotho', 'Brunei', 'New Zealand', 'Papua New
Guinea', 'Seychelles', 'Saint Kitts and Nevis', 'Dominica', 'Saint Lucia',
'Belize', 'Timor-Leste', 'Eritrea', 'Trinidad and Tobago', 'Montenegro']
```

Finding Position of India in the reversed list

In [16]:

```
position = copy.index('India')
print("\n\n\tThe postion of India in list of COVID affected countries =  {}\n\n".format(str
```

```


	The postion of India in list of COVID affected countries =  4


```

In [ ]:

```

```