In [11]:
```
# Q1.Modify the elements in the list
#a = [3,2,5,72,21,68,23,5,73,256,22,56,90]
#b) replace 90 with your name
#c) apply append operation with example
#d) apply extend operation with your own example

a = [3,2,5,72,21,68,23,5,73,256,22,56,90]

# b. replacing 90 with 'Abhi'
a[-1] = 'Abhi'
print(a)

# c apply append operation with example
a.append(905)
print(a)

# apply extend operation with your own example
b = ['joe','chandler','ross']
a.extend(b)
print(a)
```

```
[3, 2, 5, 72, 21, 68, 23, 5, 73, 256, 22, 56, 'Abhi']
[3, 2, 5, 72, 21, 68, 23, 5, 73, 256, 22, 56, 'Abhi', 905]
[3, 2, 5, 72, 21, 68, 23, 5, 73, 256, 22, 56, 'Abhi', 905, 'joe', 'chandler', 'ross']
```

In [22]:
```
#2) Create the data Frame using dictionary for 5 countries and their capital?
import pandas as pd
df =pd.DataFrame({'Country':['India', 'France', 'Japan', 'Peru', 'UK'],
                  'Capital':['Delhi', 'Paris', 'Tokyo', 'Lima', 'London']})
print(df)
```

```
  Country Capital
0   India   Delhi
1  France   Paris
2   Japan   Tokyo
3    Peru    Lima
4      UK  London
```

In [25]:
```
# 3). Create the Data Frame using List with your name, roll no, marks for three subjects?
import pandas as pd

data ={'Name':['Joe','Ross','Chandler'],
       'Roll No.':[101,102,103],
      'English':[14, 15, 15],
      'Maths':[5, 14, 15],
      'Physics':[9,15,14]}

df = pd.DataFrame(data)

print(df)
```

```
       Name  Roll No.  English  Maths  Physics
0       Joe       101       14      5        9
1      Ross       102       15     14       15
2  Chandler       103       15     15       14
```

In [48]:
```
#4)  For the given dataset using iloc and loc function access the rows and columns?

import pandas as pd

df =pd.DataFrame({'ID':[1,2,3,4,5],
                  'Name':['abhi','sai','shashank','hrishi','Ansh'],
                 'Marks':[15,15,15,15,15]})
#print(df)

# accesing data using iloc
print(df.iloc[:,1])

print('------------------------------')

# accessing data using loc
print(df.loc[:, ['Name', 'Marks']])
```

```
0        abhi
1         sai
2    shashank
3      hrishi
4        Ansh
Name: Name, dtype: object
------------------------------
       Name  Marks
0      abhi     15
1       sai     15
2  shashank     15
3    hrishi     15
4      Ansh     15
```

In [62]:
```
# 5. For the given dataset perform renaming, indexing, slicing, selecting rows and columns using conditional cr
```

```
import pandas as pd

df = pd.DataFrame({'Id':[101, 102, 103, 104],
                   "Name": ['Joe', 'Ross', 'Alice', 'Bob'],
                   'Old':[23, 24, 18, 19]})
print(df)


# renaming
df.rename(columns = {"Old":"Age"}, inplace = True)

print("\nDataFrame after renaming 'Old' to 'Age':")
print(df)

# Slicing
print('After Slicing:')
print(df.loc[[1,2],['Age', 'Id']])

# selecting rows and columns using conditional criteria
print(df[df['Age'] > 18])
```

```
    Id   Name  Old
0  101    Joe   23
1  102   Ross   24
2  103  Alice   18
3  104    Bob   19

DataFrame after renaming 'Old' to 'Age':
    Id   Name  Age
0  101    Joe   23
1  102   Ross   24
2  103  Alice   18
3  104    Bob   19
After Slicing:
   Age   Id
1   24  102
2   18  103
    Id  Name  Age
0  101   Joe   23
1  102  Ross   24
3  104   Bob   19
```

In [63]:
```
# 6q. Create the data frame with the null values and show how you handle null values

import pandas as pd
import numpy as np

df = pd.DataFrame({
    'A':[1,2,np.nan],
    'B':[5,np.nan, np.nan],
    'C':[1,2,3]
})

print(df)
```

```
     A    B  C
0  1.0  5.0  1
1  2.0  NaN  2
2  NaN  NaN  3
```

In [64]:
```
# Handling missing data

# using fillna() to fill the missing places with 0 or we also fill with mean 'fillna(df.mean)'
df.fillna(0)
```

Out[64]:

	A	B	C
0	1.0	5.0	1
1	2.0	0.0	2
2	0.0	0.0	3

In [65]:
```
# We can also drop row or column with missing data
# dropping of row with null values

df.dropna(axis=0)
```

Out[65]:

	A	B	C
0	1.0	5.0	1

In [66]:
```
# dropping of column with null values
df.dropna(axis=1)
```

Out[66]:

	C
0	1
1	2
2	3

In [ ]:
```
#7. merge the two data Frames  using different join and function argument
```

In [68]:
```
import pandas as pd

# Create DataFrame df1
df1 = pd.DataFrame({
    'A': ['A0', 'A1', 'A2'],
    'B': ['B0', 'B1', 'B2'],
    'key': ['K0', 'K1', 'K2']
})

# Create DataFrame df2
df2 = pd.DataFrame({
    'C': ['C0', 'C1', 'C2'],
    'D': ['D0', 'D1', 'D2'],
    'key': ['K0', 'K2', 'K3']
})
```

In [70]:
```
# inner join
f_inner = pd.merge(df1, df2, on='key', how='inner')
print(f_inner)
```

```
    A   B key   C   D
0  A0  B0  K0  C0  D0
1  A2  B2  K2  C1  D1
```

In [71]:
```
# Outer join
f_outer = pd.merge(df1, df2, on='key', how='outer')
print(f_outer)
```

```
     A    B key    C    D
0   A0   B0  K0   C0   D0
1   A1   B1  K1  NaN  NaN
2   A2   B2  K2   C1   D1
3  NaN  NaN  K3   C2   D2
```

In [72]:
```
# left join
left = pd.merge(df1, df2, on='key', how='left')
print(left)
```

```
    A   B key    C    D
0  A0  B0  K0   C0   D0
1  A1  B1  K1  NaN  NaN
2  A2  B2  K2   C1   D1
```

In [74]:
```
# right join
right = pd.merge(df1, df2, on='key', how='right')
print(right)
```

```
     A    B key   C   D
0   A0   B0  K0  C0  D0
1   A2   B2  K2  C1  D1
2  NaN  NaN  K3  C2  D2
```

Question 8

In [76]:
```
# 8)concat the 2 data Frame row wise and column wise?

df1 = pd.DataFrame({
    'A': ['A0', 'A1', 'A2'],
    'B': ['B0', 'B1', 'B2']
})

# Create DataFrame df2
df2 = pd.DataFrame({
    'A': ['A3', 'A4', 'A5'],
    'B': ['B3', 'B4', 'B5']
})

concat = pd.concat([df1, df2])
print(concat)
```

```
    A   B
0  A0  B0
1  A1  B1
2  A2  B2
0  A3  B3
1  A4  B4
2  A5  B5
```

In [78]:
```
#9.) add the new column to the existing  Data frame using your own example?

import pandas as pd

# Create DataFrame df
df = pd.DataFrame({
    'A': ['A0', 'A1', 'A2'],
    'B': ['B0', 'B1', 'B2']
})

print(df)

# Add new column 'C'
print("After Adding One More Column")
df['C'] = ['C0', 'C1', 'C2']

print(df)
```

```
    A   B
0  A0  B0
1  A1  B1
2  A2  B2
After Adding One More Column
    A   B   C
0  A0  B0  C0
1  A1  B1  C1
2  A2  B2  C2
```

In [ ]:

In [ ]:

In [ ]:

In [ ]: