In [1]:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
```

In [2]:

```
Item = pd.read_csv("Item.csv")
print(Item)
```

```
            Item       Description  Retail_Price  Loyalty_Discount  CustomerID
0     8619953091         Pillowcase         18.96              0.02   200000663
1     2020397001   Men's Pajama Set         23.10              0.00   200000661
2     4681342313             Sheets         29.90              0.10   200000662
3     6697166886               Coat        159.80              0.07   100000007
4     6697166886               Coat        159.80              0.06   400000180
...          ...                ...           ...               ...         ...
3450  4619440506             Shorts         69.75              0.10   200000662
3451  5153370805             Shorts         43.96              0.02   100000854
3452  3123824581         Sweatpants         43.89              0.06   200000793
3453  9195451761         Hand Towel         51.66              0.07   200000263
3454  6660530324          Washcloth         35.14              0.05   200000670

[3455 rows x 5 columns]
```

In [3]:

```
Item.head()
```

Out[3]:

	Item	Description	Retail_Price	Loyalty_Discount	CustomerID
0	8619953091	Pillowcase	18.96	0.02	200000663
1	2020397001	Men's Pajama Set	23.10	0.00	200000661
2	4681342313	Sheets	29.90	0.10	200000662
3	6697166886	Coat	159.80	0.07	100000007
4	6697166886	Coat	159.80	0.06	400000180

In [4]:

```
Item.loc()
```

Out[4]:

```
<pandas.core.indexing._LocIndexer at 0x243022b5c20>
```

In [5]:

```
Item.tail()
```

Out[5]:

	Item	Description	Retail_Price	Loyalty_Discount	CustomerID
3450	4619440506	Shorts	69.75	0.10	200000662
3451	5153370805	Shorts	43.96	0.02	100000854
3452	3123824581	Sweatpants	43.89	0.06	200000793
3453	9195451761	Hand Towel	51.66	0.07	200000263
3454	6660530324	Washcloth	35.14	0.05	200000670

In [6]:

```
Item.shape
```

Out[6]:

```
(3455, 5)
```

In [7]:

```
Item.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 3455 entries, 0 to 3454
Data columns (total 5 columns):
 #   Column            Non-Null Count  Dtype  
---  ------            --------------  -----  
 0   Item              3455 non-null   int64  
 1   Description       3455 non-null   object 
 2   Retail_Price      3455 non-null   float64
 3   Loyalty_Discount  3455 non-null   float64
 4   CustomerID        3455 non-null   int64  
dtypes: float64(2), int64(2), object(1)
memory usage: 135.1+ KB
```

In [8]:
```
Item.nunique() ## dtype int64 means data is stored as integer 64 bytes in python
```

Out[8]:
```
Item                126
Description          68
Retail_Price        110
Loyalty_Discount     11
CustomerID          942
dtype: int64
```

In [9]:
```
Item.columns
```

Out[9]:
```
Index(['Item', 'Description', 'Retail_Price', 'Loyalty_Discount',
       'CustomerID'],
      dtype='object')
```

In [10]:
```
Item.isnull().sum()
```

Out[10]:
```
Item                0
Description         0
Retail_Price        0
Loyalty_Discount    0
CustomerID          0
dtype: int64
```

In [11]:
```
Item.notnull().min()
```

Out[11]:
```
Item                True
Description         True
Retail_Price        True
Loyalty_Discount    True
CustomerID          True
dtype: bool
```

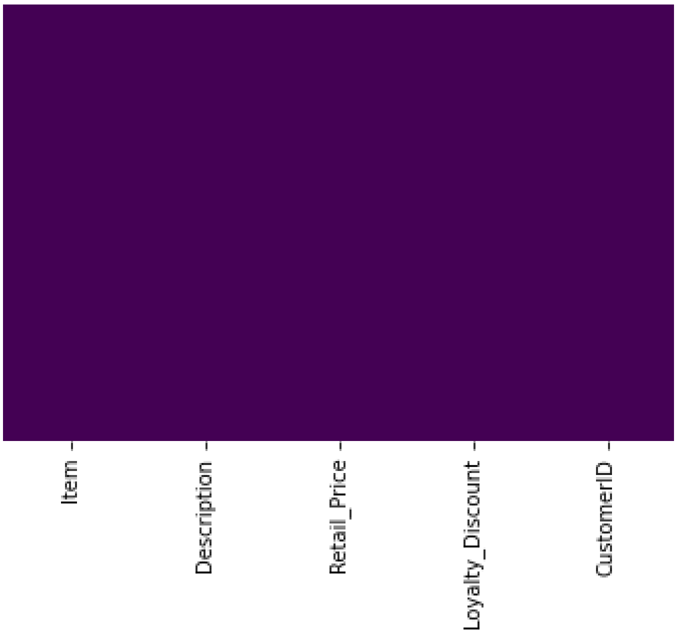
In [12]:
```
Item.describe()
```

Out[12]:

	Item	Retail_Price	Loyalty_Discount	CustomerID
count	3.455000e+03	3455.000000	3455.000000	3.455000e+03
mean	5.276712e+09	58.526237	0.050457	1.797979e+08
std	2.600486e+09	34.464217	0.032215	9.563412e+07
min	1.039855e+09	5.600000	0.000000	1.000000e+08
25%	2.963301e+09	31.800000	0.020000	1.000003e+08
50%	5.145202e+09	51.660000	0.050000	1.000009e+08
75%	7.645689e+09	79.800000	0.080000	2.000009e+08
max	9.916068e+09	159.800000	0.100000	4.000009e+08

In [13]:
```
sns.heatmap(Item.isnull(),yticklabels= False, cbar = False, cmap='viridis')
plt.figure(figsize = (15,10))
```

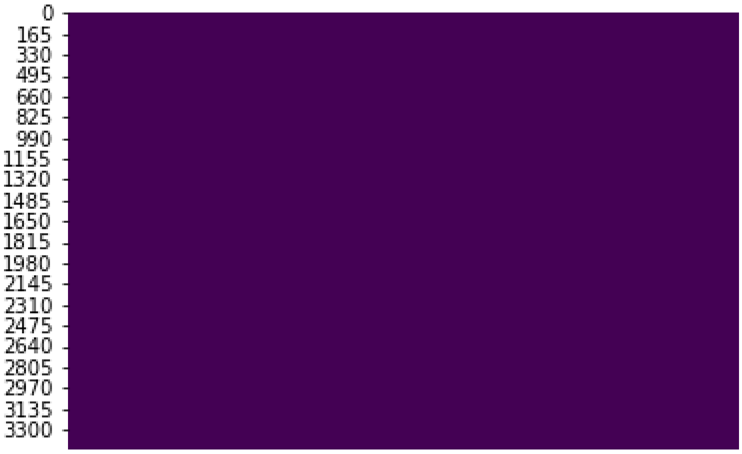
Out[13]:
```
<Figure size 1080x720 with 0 Axes>
```



```
<Figure size 1080x720 with 0 Axes>
```

In [14]:
```
sns.heatmap(Item.notnull(),xticklabels= False, cbar = False, cmap='viridis')
plt.figure(figsize = (15,10))
```

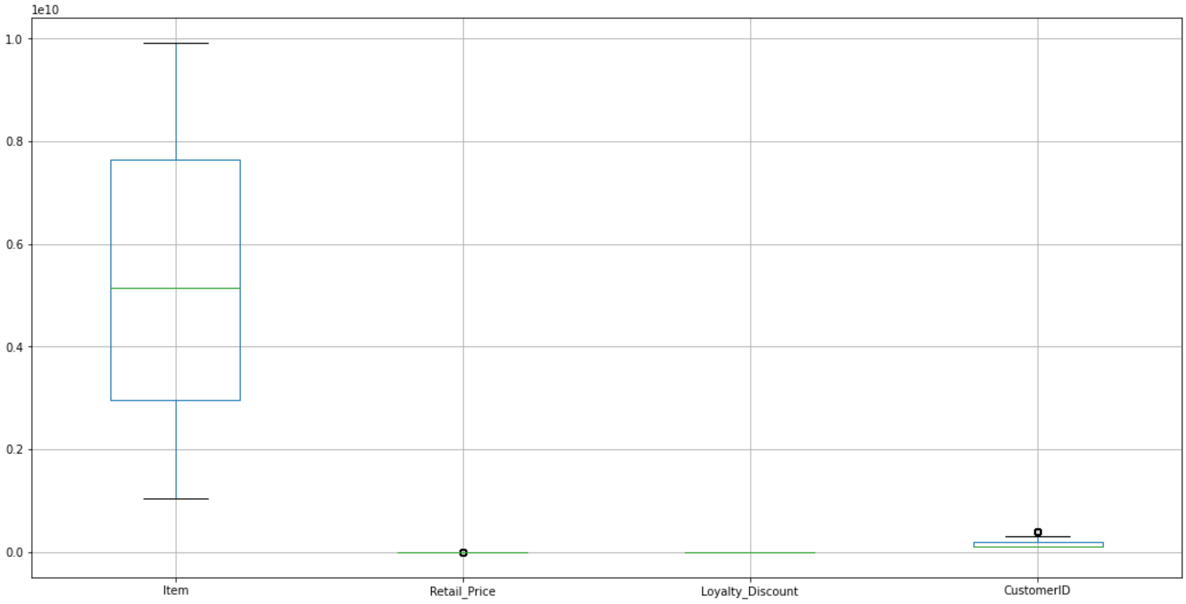
Out[14]:
```
<Figure size 1080x720 with 0 Axes>
```



```
<Figure size 1080x720 with 0 Axes>
```

In [15]:
```
Item.boxplot(figsize= (18,9))
```

Out[15]:
```
<AxesSubplot:>
```



In [16]:
```
data = input("enter the date of birth in the format DD/MMM/YYYY:")
                                                   ##012345678910


date = data[0:2]
month = data[3:6]
year = data[7:]

print(f"The user was born in the month of: {month} in the year: {year} on date: {da
```

```
enter the date of birth in the format DD/MMM/YYYY:23/07/1994
The user was born in the month of: 07/ in the year: 994 on date: 23
```

palindrome

data = input("input a string") reverse = data[::-1]

print(data , reverse)

In [17]:
```
data = input("input a string: ")
reverse = data[::-1]

print(data , reverse)
```

```
input a string: mum
mum mum
```

In [18]:
```
data = input("input a string: ")
reverse = data[::-1]

if data == reverse:
    print(f"yes input string :{data} is a palindrome")
else:
     print(f"no input string :{data} is Not a palindrome")
```

```
input a string: mum
yes input string :mum is a palindrome
```

In [19]:
```
data = input ("input a string: ")
reverse = data[::-1]

if data == reverse:
    print(f"yes input string :{data} is a palindrome")
```

```
else:
     print(f"no input string :{data} is Not a palindrome")
```

```
input a string: root
no input string :root is Not a palindrome
```

In [20]:

```
data = input ("input a string: ")
reverse = data[::-1]

if data == reverse:
    print(f"yes input string :{data} is a palindrome")
else:
     print(f"no input string :{data} is Not a palindrome")
```

```
input a string: jyoti
no input string :jyoti is Not a palindrome
```

Mutablitity

mutable means which is subject to change which is changable

changes are not allowed in strings it can be done through concatination

replace s with f name==Sunny

name = "Sunny" name[0]= "F"

print(name)

In [21]:

```
name = "Sunny"
name[0]= "F"

print(name)
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Input In [21], in <cell line: 2>()
      1 name = "Sunny"
----> 2 name[0]= "F"
      4 print(name)

TypeError: 'str' object does not support item assignment
```

concatination

name = "Sunny" "F" + name[1:]

In [ ]:

```
name = "Sunny"
"F" + name[1:]
```

replace

name = "Sunny" name = name.replace('S','F') print(name)

In [ ]:
```
name = "Sunny"
name = name.replace('S','F')
print(name)
```

In [ ]:
```
name = "Jyoti VERMA"

name = name.replace('Jyoti', 'JYOTI')
print(name)
```

# string indexing string slicing string skipping string reversal string replace method length of the string string is immutable bcz we cannot assign any value directly using indexing

In [ ]:
```
name = "Jyoti Verma"

name[6]
```

In [ ]:
```
name = "Jyoti Verma"
name[6:]
```

In [ ]:
```
name = "Jyoti Verma"
name[::2]
```

In [ ]:
```
name = "Jyoti Verma"
name[::-1]
```

In [ ]:
```
name = "Jyoti Verma"

name=name.replace('Jyoti Verma','JYOTI VERMA')
print(name)
```

In [ ]:
```
name = "JYOTIVERMA"
len(name)
```

In [ ]:
```
name= "Jyoti Verma"

name.lower()
```

In [ ]:
```
name.upper()
```

In [ ]:
```
name.title()
```

upper, lower, tittle, capitalize and swapcase cases

In [ ]:
```
name= input("Enter your Name: ")
print(f"Name in upper case :{name.upper()}")
print(f"Name in lower case :{name.lower()}")
print(f"Name in title case :{name.title()}")
```

In [ ]:
```
name= input("Enter your Name: ")
print(f"Name in upper case :{name.upper()}")
print(f"Name in lower case :{name.lower()}")
print(f"Name in title case :{name.title()}")
print(f"Name in capitalize case :{name.capitalize()}")
print(f"Name in swapcase case :{name.swapcase()}")
```

Join Function

In [ ]:
```
name= input("Enter your Name: ")
"*".join(name)
```

In [ ]:
```
name = "Anchal Verma"
"|".join(name)
```

In [ ]:
```
name = "Anchal Verma"
"".join(reversed(name))
```

we cannot use replace in join as in replace changes are made after word

In [ ]:
```
name = "Anchal Verma"
name.replace(" ", ",")
```

to remove extra spaces

In [ ]:
```
name = "Anchal Verma    "
name.strip(" ")
```

In [ ]:
```
name = "    Anchal Verma    "
name.rstrip(" ")
```

In [ ]:
```
name = "    Anchal Verma    "
name.lstrip(" ")
```

In [ ]:
```
## another example
```

In [ ]:
```
name = "    Anchal Verma    "
name.replace(" ", "")
```

In [ ]:
```
## formatting
```

In [ ]:
```
name = "Anchal Verma"
name.center(20,"+")
```

In [ ]:
```
## Is upper, Is lower, Is space, Is title,
```

In [ ]:
```
name = "Anchal Verma"
name.islower()
```

In [ ]:

```
name = "Anchal Verma"
name.isupper()
```

In [ ]:

```
name = "Anchal Verma"
name.isspace()
```

In [ ]:

```
name = "Anchal Verma"
name.istitle()
```

In [ ]:

```
name = input("Enter your Name: ")

print(f"user input: {name}")

if name.istitle():
    print(f"user has given correct input")
else:
    print(f"Wrong input we need to autocorrect it")
    name = name.title()
    print(f"Correct output: {name}")
```

In [ ]:

```
name = input("Enter your Name: ")

print(f"user input: {name}")

if name.istitle():
    print(f"user has given correct input")
elif name.isspace():
    print(f"wrong input please try again!")
else:
    print(f"Wrong input we need to autocorrect it")
    name = name.title()
    print(f"Correct output: {name}")
```

In [ ]:

```
name = input("Enter your Name: ")

print(f"user input: {name}")

if name.istitle():
    print(f"user has given correct input")
elif name.isspace():
    print(f"wrong input please try again!")
else:
    print(f"Wrong input we need to autocorrect it")
    name = name.title()
    print(f"Correct output: {name}")
```

In [ ]:

```
phone_number = "9887676755"
phone_number.isdigit()
```

In [ ]:

```
phone_number = "98876767fshfgy"
phone_number.isdigit()
```

In [ ]:

```
phone_number = input(f"Enter your number: ")

if phone_number.isdigit() and len(phone_number) == 10:
    print(f"user input is correct")
else:
    print(f"invalid input")
```

In [ ]:

```
phone_number = input(f"Enter your number: ")

if phone_number.isdigit() and len(phone_number) == 10:
    print(f"user input is correct")
else:
    print(f"invalid input")
```

In [ ]:

```
phone_number = input(f"Enter your number: ")

if phone_number.isdigit() and len(phone_number) == 10:
    print(f"user input is correct")
else:
    print(f"invalid input")
```

In [ ]:

```
phone_number = input(f"Enter your number: ")

if phone_number.isdigit() and len(phone_number) == 10 and phone_number!= "000000000
    print(f"user input is correct")
else:
    print(f"invalid input")
```

In [ ]:

```
phone_number = "0000000000"
phone_number.startswith("0")
```

In [ ]:

```
phone_number = "0000000000"
phone_number.endswith("0")
```

In [ ]:

```
phone_number = "+919773870799"
phone_number.startswith("+91")
```

In [ ]:

```
phone_number[1:].isdigit()
```

In [ ]:

```
len(phone_number)
```

In [ ]:

```
phone_number = input(f"Enter your India Phone Number: ")

if phone_number.startswith("+91") and phone_number[1:].isdigit() and len(phone_num
    print(f"User input is correct")

else:
      print(f"Invalid input")
```

if elif and else condition

In [ ]:

```
total_amount = 100 + 90 + 100 + 1000
print(f"Cart Total : {total_amount}")

if total_amount > 999:
    price_after_discount = total_amount * 70/100
    print(f"Pay Amount : {price_after_discount}")
elif total_amount <= 999 and total_amount >= 499:
    price_after_discount = total_amount * 80/100
    print(f"Pay Amount : {price_after_discount}")
```

```
else:
    print(f"Pay Amount : {total_amount}")
```

In [ ]:

```
total_amount = 100 + 90 + 100
print(f"Cart Total : {total_amount}")

if total_amount > 999:
    price_after_discount = total_amount * 70/100
    print(f"Pay Amount : {price_after_discount}")
elif total_amount <= 999 and total_amount >= 499:
    price_after_discount = total_amount * 80/100
    print(f"Pay Amount : {price_after_discount}")
else:
    print(f"Pay Amount : {total_amount}")
```

In [ ]:

```
total_amount = 100 + 90 + 500
print(f"Cart Total : {total_amount}")

if total_amount > 999:
    price_after_discount = total_amount * 70/100
    print(f"Pay Amount : {price_after_discount}")
elif total_amount <= 999 and total_amount >= 499:
    price_after_discount = total_amount * 80/100
    print(f"Pay Amount : {price_after_discount}")
else:
    print(f"Pay Amount : {total_amount}")
```

In [ ]:

```
total_amount = 100 + 90 + 100 + 1000
print(f"Cart Total : {total_amount}")

if total_amount > 1499:
    price_after_discount = total_amount * 60/100
    print(f"Pay Amount : {price_after_discount}")
elif total_amount <= 1499 and total_amount >= 999:
    price_after_discount = total_amount * 70/100
    print(f"Pay Amount : {price_after_discount}")
elif total_amount <= 999 and total_amount >= 499:
    price_after_discount = total_amount * 80/100
    print(f"Pay Amount : {price_after_discount}")
else:
    print(f"Pay Amount : {total_amount}")
```

In [ ]:

```
total_amount = 100 + 90 + 400 + 1000
print(f"Cart Total : {total_amount}")

if total_amount > 1499:
    price_after_discount = total_amount * 60/100
    print(f"Pay Amount : {price_after_discount}")
elif total_amount <= 1499 and total_amount >= 999:
    price_after_discount = total_amount * 70/100
    print(f"Pay Amount : {price_after_discount}")
elif total_amount <= 999 and total_amount >= 499:
    price_after_discount = total_amount * 80/100
    print(f"Pay Amount : {price_after_discount}")
else:
    print(f"Pay Amount : {total_amount}")
```

nested if else

In [ ]:
```
value = float(input(f"Enter a Number: "))
if value >=0:
    if value == 0:
        print("Its zero")
    else:
        print("Its a Positive Number")
else:
     print("Its a Negative Number")
```

In [ ]:
```
value = float(input(f"Enter a Number: "))
if value >=0:
    if value == 0:
        print("Its zero")
    else:
        print("Its a Positive Number")
else:
     print("Its a Negative Number")
```

In [ ]:
```
value = float(input(f"Enter a Number: "))
if value >=0:
    if value == 0:
        print("Its zero")
    else:
        print("Its a Positive Number")
else:
     print("Its a Negative Number")
```

single line if else condition

In [ ]:
```
value = float(input(f"Enter a Number: "))
if value > 99: print("Yes")
else: print("NO")
```

In [ ]:
```
value = float(input(f"Enter a Number: "))
if value > 99: print("Yes")
else: print("NO")
```

single line codes are not neat always go for different lines of code

Loops 1.

1. while loop
2. For loop
3. Loop control: .break .continue .pass 4 Nested loop

In [ ]:
```
total_marks = 1000
cutoff = 400
scores = [100, 200, 300, 399, 500]
year= 0
```

```
while scores[year]< cutoff:
    print(f"Your score is: {scores[year]} , cutoff : {cutoff}")
    print(f"I will attemt next year")
    year = year+1
```

Infinete loop

if the condition is always true then your loop will keep on running

avoid such a condition

In [ ]:

```
# kid is counting 10 rupee note

notes = 5
i=1 #1st note

while i <= notes:
    print(f"current sum: {i*10}")
    i=i+1
```

In [ ]:

```
notes = 10
i=1 #1st note

while i <= notes:
    print(f"current sum: {i*10}")
    i=i+1
```

In [ ]:

```
station = ["station01", "station02", "station03", "station04"]

current _station = 0
destination_station = "station03"

while station[current_station] != destination_station:
    print(f"current station is : {station[current_station]}")
    print(f"My destination station: {destination_station}")
    print(f"Continue the journey I haven't reached the station")
    current_station = current_station + 1
    print(f"Next station is: {station[current_station]}")
    print("----------")
```

In [ ]:

```
station = ["station01", "station02", "station03", "station04"]

current_station = 0
destination_station = "station03"

while station[current_station] != destination_station:
    print(f"current station is : {station[current_station]}")
    print(f"My destination station: {destination_station}")
    print(f"Continue the journey I haven't reached the station")
    current_station = current_station + 1
    print(f"Next station is: {station[current_station]}")
    print("----------")
```

```
else:
    print(f"I have arrived at: {station[current_station]}" )
```

In [ ]:

```
notes = 10
i=1 #1st note

while i <= notes:
    print(f"Condition: {i <= notes}" )
    print(f"current sum: {i*10}")
    i=i+1
    print('----------')
else:
    print(f"Condition: {i <= notes}" )
    print('No more 10 rupees notes')
```

For loop

In [ ]:

```
for i in range (0,10): #it is no not printing 10 times 10 as 10 here is out of bou
    print(f"10 x {i} = {10*i}")
```

In [ ]:

```
for i in range (0,10):
    print(f"10 x {i+1} = {10*(i+1)}")
```

In [ ]:

```
for i in range (0,11):
    print(f"10 x {i} = {10*i}")
```

In [ ]:

```
cost_of_items = [100, 200, 129, 456]

total_sum = 0

for i in range (0,4):
    print(f"{total_sum} = {total_sum} + {cost_of_items[i]}")
    total_sum = total_sum + cost_of_items[i]
    print(f"After adding total_sum = {total_sum}")
    print("--------")
print(f"Total amount to be paid : {total_sum}")
```

if you donot want to mention the range

In [ ]:

```
cost_of_items = [100, 200, 129, 456]

total_sum = 0
N= len(cost_of_items)

for i in range (0,N):
    print(f"{total_sum} = {total_sum} + {cost_of_items[i]}")
    total_sum = total_sum + cost_of_items[i]
    print(f"After adding total_sum = {total_sum}")
    print("--------")
print(f"Total amount to be paid : {total_sum}")
```

In [ ]:

```
cost_of_items = [100, 200, 129, 456, 1000]

total_sum = 0
N= len(cost_of_items)
```

```
for i in range (0,N):
    print(f"{total_sum} = {total_sum} + {cost_of_items[i]}")
    total_sum = total_sum + cost_of_items[i]
    print(f"After adding total_sum = {total_sum}")
    print("--------")
print(f"Total amount to be paid : {total_sum}")
```

Another method

In [ ]:

```
cost_of_items = [100, 200, 129, 456, 1000]

total_sum = 0

for cost in cost_of_items:
    print(f"{total_sum} = {total_sum} + {cost}")
    total_sum = total_sum + cost
    print(f"After adding total_sum = {total_sum}")
    print("--------")
print(f"Total amount to be paid : {total_sum}")
```

In [ ]:

```
station = ["station01", "station02", "station03", "station04"]

current_station = 0
destination_station = "station03"
for current_station in station:
    if current_station == "station02":
        continue
    print(f"Current station is: {current_station}")
```

In [ ]:

```
station = ["station01", "station02", "station03", "station04"]

current_station = 0
destination_station = "station03"
for current_station in station:
    if current_station == "station02":
        break
    print(f"Current station is: {current_station}")
```

In [ ]:

```
for i in range (1, 20):
    if i%2==0:
        continue
    print(i)
```

In [ ]:

```
for i in range (1, 20):
    if i%2!=0:
        continue
    print(i)
```

In [ ]:

```
for i in range (1, 100):
    print(i)
    if i > 50:
        break
```

In [ ]:

```
for i in range (1, 100):
    if i > 50:
        break
    print(i)
```

In [ ]:

```
num = 80

if num > 100:
    print(f"The Number is Greater than 100")
elif num > 80 and num <= 100:
    print(f"The Number is in the range 80 to 100")
elif num > 60 and num <= 80:
    print(f"The Number is in the range 80 to 100")
elif num > 40 and num <= 60:
    print(f"The Number is in the range 80 to 100")
else:
    print(f"The Number is Less than 40")
```

In [ ]:

```
num = 70

if num > 100:
    print(f"The Number is Greater than 100, Grade: A")
elif num > 80 and num <= 100:
    print(f"The Number is in the range 80 to 100, Grade: B")
elif num > 60 and num <= 80:
    print(f"The Number is in the range 80 to 100, Grade: C")
elif num > 40 and num <= 60:
    print(f"The Number is in the range 80 to 100, Grade: D")
else:
    print(f"The Number is Less than 40")
```

LIST

In [22]:

```
list_of_item_to_purchase = list()

N = int(input("No of item to be purchased: "))
for i in range (N):
    user_input = input("Enter the item that you want to purchase: ")
    list_of_item_to_purchase = list_of_item_to_purchase + [user_input]
print(list_of_item_to_purchase)
```

```
No of item to be purchased: 3
Enter the item that you want to purchase: milk
Enter the item that you want to purchase: bread
Enter the item that you want to purchase: wheat
['milk', 'bread', 'wheat']
```

In [23]:

```
list_of_item_to_purchase = list()

N = int(input("No of item to be purchased: "))
i = 0
while i < N:
    user_input = input("Enter the item that you want to purchase: ")
    list_of_item_to_purchase = list_of_item_to_purchase + [user_input]
    i = i+1
print(list_of_item_to_purchase)
```

```
No of item to be purchased: 3
Enter the item that you want to purchase: milk
Enter the item that you want to purchase: bread
Enter the item that you want to purchase: pen
['milk ', 'bread', 'pen']
```

In [2]:

```
list_of_item_to_purchase = ["pen", "papper", "book", "marker", "glue", "scissors"
print(list_of_item_to_purchase)
```

```
['pen', 'papper', 'book', 'marker', 'glue', 'scissors']
```

In [3]:

```
new_item = input("Anything else to be added: ")

if new_item in list_of_item_to_purchase:
    print("YES")
else:
    print("Adding item to the list")
    list_of_item_to_purchase = list_of_item_to_purchase +[new_item]
```

```
Anything else to be added: table
Adding item to the list
```

In [4]:

```
list_of_item_to_purchase
```

Out[4]:

```
['pen', 'papper', 'book', 'marker', 'glue', 'scissors', 'table']
```

In [5]:

```
new_item = input("Anything else to be added: ")

if new_item not in list_of_item_to_purchase:
    print("NO")
else:
    print("Adding item to the list")
    list_of_item_to_purchase = list_of_item_to_purchase +[new_item]
```

```
Anything else to be added: Notebook
NO
```

In [6]:

```
list_of_item_to_purchase
```

Out[6]:

```
['pen', 'papper', 'book', 'marker', 'glue', 'scissors', 'table']
```

In [7]:

```
new_item = input("Anything else to be added: ")

if new_item not in list_of_item_to_purchase:
    print("NO Adding item to the list")
    list_of_item_to_purchase = list_of_item_to_purchase +[new_item]

list_of_item_to_purchase
```

```
Anything else to be added: notebook
NO Adding item to the list
```

Out[7]:

```
['pen', 'papper', 'book', 'marker', 'glue', 'scissors', 'table', 'notebook']
```

MAX & MIN

In [9]:

```
For_max = [1,99,38,9900,28,7]
max(For_max)
```

Out[9]:

```
9900
```

In [10]:

```
For_min = [1,99,38,9900,28,7]
min(For_min)
```

Out[10]:

```
1
```

In [12]:
```
str_max = ["A","a","b","e","d"]
max(str_max)
```
Out[12]:
```
'e'
```

In [13]:
```
str_min = ["A","a","b","e","d"]  ##to see the comparision look for ASCII code amer
min(str_min)
```
Out[13]:
```
'A'
```

In [14]:
```
str_max = ["A","a","b","e","d", 6, 900, 67] ##string and integer cannot be compare
max(str_max)
```
```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Input In [14], in <cell line: 2>()
      1 str_max = ["A","a","b","e","d", 6, 900, 67]
----> 2 max(str_max)

TypeError: '>' not supported between instances of 'int' and 'str'
```

In [16]:
```
str_max = [1, 2.5, 6.8, 88, 90.65]
max(str_max)
```
Out[16]:
```
90.65
```

In [17]:
```
For_min = [-1,99,38,99,28,-7]
min(For_min)
```
Out[17]:
```
-7
```

In [19]:
```
str_max = [1, 2*50, 6.8, 88, 90]
max(str_max)
```
Out[19]:
```
100
```

In [20]:
```
str_var_max = ["jyoti","anchal","dew","zane","aisha"]
max(str_var_max)  ###answer is zane as it starts with z
```
Out[20]:
```
'zane'
```

In [21]:
```
str_var_max = ["jyoti","anchal","dew","zane","zuric"]
max(str_var_max)  ###answer is zane as it starts with z and next value is u next h
```
Out[21]:
```
'zuric'
```

In [22]:
```
str_example = ["jyoti","anchal","dew","zane","aisha"]
print(str_example)
```
```
['jyoti', 'anchal', 'dew', 'zane', 'aisha']
```

In [23]:
```
max_len = 0
result = ""
for example in str_example:
    print(example , len(example))
    if len(example) > max_len:
        max_len = len(example)
        result = example
print(f"Result: {max_len}, name: {result}")
```

```
jyoti 5
anchal 6
dew 3
zane 4
aisha 5
Result: 6, name: anchal
```

APPEND

In [24]:

```
A = [10,12,45,78]
print(A)
```

```
[10, 12, 45, 78]
```

In [26]:

```
A.append("python")
print(A)
```

```
[10, 12, 45, 78, 'python', 'python']
```

In [27]:

```
A= A + ["jupiter"]
print(A)
```

```
[10, 12, 45, 78, 'python', 'python', 'jupiter']
```

In [29]:

```
list_of_item_to_purchase = list()

N = int(input("No of item to be purchased: "))
for i in range (N):
    user_input = input("Enter the item that you want to purchase: ")
    list_of_item_to_purchase.append(user_input)
print(list_of_item_to_purchase)
```

```
No of item to be purchased: 2
Enter the item that you want to purchase: tea
Enter the item that you want to purchase: coffee
['tea', 'coffee']
```

POP

In [30]:

```
A = [10,12,45,78]

A.pop()
```

Out[30]:

```
78
```

In [31]:

```
print(A)
```

```
[10, 12, 45]
```

In [32]:

```
A = [10,12,45,78]

A.pop(0)
print(A)
```

```
[12, 45, 78]
```

In [33]:

```
A = [1, 34, 56, 76, 99]

for inx in range(len(A)):
    num = A[inx]
    print(inx , num)
    if num % 2 != 0:
        print(f"odd num: {num} ")
    else:
        print(f"even num: {num}")
```

```
0 1
odd num: 1
1 34
even num: 34
2 56
even num: 56
3 76
even num: 76
4 99
odd num: 99
```

In [35]:

```
A = [1, 34, 56, 76, 99]

for inx in range(len(A)):
    num = A[inx]
    print(inx , num)
    if num % 2 != 0:
        print(f"odd num: {num} ")
    else:
        print(f"even num: {num}")
```

```
0 1
odd num: 1
1 34
even num: 34
2 56
even num: 56
3 76
even num: 76
4 99
odd num: 99
```

REVERSE ASC/DES

In [36]:

```
A = [10, 88, 65, 56, 77]

A.sort()
```

In [40]:

```
print(A)
```

```
[10, 23, 45, 56, 77]
```

In [39]:

```
sorted(A)
```

Out[39]:

```
[10, 23, 45, 56, 77]
```

In [41]:

```
A[::-1]
```

Out[41]:

```
[77, 56, 45, 23, 10]
```

In [42]:

```
A.reverse()
```

In [44]:

```
print(A)
```

```
[77, 56, 45, 23, 10]
```

In [45]:

```
A = [10, 45, 77, 88, 3]

A.reverse()
```

In [46]:

```
print(A)
```

```
[3, 88, 77, 45, 10]
```

NESTING

In [47]:

```
A = [[10, 14, 65], [77, 89, 12], [34, 89, 90]]
print(A)
```

```
[[10, 14, 65], [77, 89, 12], [34, 89, 90]]
```

In [48]:

```
A[0]
```

Out[48]:

```
[10, 14, 65]
```

In [49]:

```
A[1]
```

Out[49]:

```
[77, 89, 12]
```

In [50]:

```
A[1][1]
```

Out[50]:

```
89
```

In [ ]:

```
A = [[10, 14, 65], [77, 89, 12, [500, 789]], [34, 89, 90]]
```

In [51]:

```
A[1][1]
```

Out[51]:

```
89
```

In [53]:

```
A[1]
```

Out[53]:

```
[77, 89, 12]
```

In [54]:

```
A = [[10, 14, 65], [77, 89, 12, [500, 789]], [34, 89, 90]]
print(A)
```

```
[[10, 14, 65], [77, 89, 12, [500, 789]], [34, 89, 90]]
```

In [55]:

```
A[1]
```

Out[55]:

```
[77, 89, 12, [500, 789]]
```

In [56]:

```
A[1][3][0]
```

Out[56]:

```
500
```

In [57]:

```
A =[[1, 2, 3], [11, 22, 33, [500, 502]], [45, [239, "HII"], 56, 77]]
print(A)
```

```
[[1, 2, 3], [11, 22, 33, [500, 502]], [45, [239, 'HII'], 56, 77]]
```

In [58]:

```
A[2][1][0]
```

```
Out[58]: 239
```

```
In [59]: A[2][1][1]
```

```
Out[59]: 'HII'
```

```
In [60]: A[-1][1][-1]
```

```
Out[60]: 'HII'
```

```
In [61]: amazon_cart = [["watch", 5000],["phone",10000],["laptop", 50000]]

         print(amazon_cart)
```

```
[['watch', 5000], ['phone', 10000], ['laptop', 50000]]
```

```
In [62]: amazon_cart[0][1]
```

```
Out[62]: 5000
```

```
In [63]: amazon_cart[1][1]
```

```
Out[63]: 10000
```

```
In [64]: amazon_cart[2][1]
```

```
Out[64]: 50000
```

```
In [65]: amazon_cart[0][1]+amazon_cart[1][1]+amazon_cart[2][1]
```

```
Out[65]: 65000
```

```
In [66]: total_cost =0
         for i in range(len(amazon_cart)):
             print(amazon_cart[i][1])
             total_cost = total_cost + amazon_cart[i][1]

         print(total_cost)
```

```
5000
10000
50000
65000
```

```
In [67]: total_cost =0
         for item in amazon_cart:
             print(item[1])
             total_cost = total_cost + item[1]

         print(total_cost)
```

```
5000
10000
50000
65000
```

```
In [68]: total_cost =0
         print(f"empty cart: {total_cost}")

         for item in amazon_cart:
             print(item[1])
```

```
    total_cost = total_cost + item[1]
    print(f"cart after adding {item[0]} : {total_cost}")

print(total_cost)
```

```
empty cart: 0
5000
cart after adding watch : 5000
10000
cart after adding phone : 15000
50000
cart after adding laptop : 65000
65000
```

List Comprehension

list which contain list of square of no. between 1 to 10

In [69]:

```
A = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
print(A)
```

```
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
```

In [70]:

```
ans = list()

for element in A:
    print(element**2)
    ans.append(element**2)
print(ans)
```

```
1
4
9
16
25
36
49
64
81
100
[1, 4, 9, 16, 25, 36, 49, 64, 81, 100]
```

In [71]:

```
ans = [element**2 for element in A]

print(ans)
```

```
[1, 4, 9, 16, 25, 36, 49, 64, 81, 100]
```

In [73]:

```
A = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]

ans = list()
for element in A:
    if element % 2 != 0:
        print(element**2)
        ans.append(element**2)
```

```
1
9
25
49
81
```

In [76]:

```
A = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]

ans = list()
for element in A:
    if element % 2 != 0:
        ans.append(element**2)
print(ans)
```

```
[1, 9, 25, 49, 81]
```

In [77]:

```
A = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]

ans = list()
for element in A:
    if element % 2 != 0:
        print(element**2)
        ans.append(element**2)
print(ans)
```

```
1
9
25
49
81
[1, 9, 25, 49, 81]
```

In [80]:

```
ans = [element**2 for element in A if element %2 != 0]

print(ans)
```

```
[1, 9, 25, 49, 81]
```

In [81]:

```
A = [1, 12, 12, 12, 5, 6, 7, ]

print(A)
```

```
[1, 12, 12, 12, 5, 6, 7]
```

In [82]:

```
A.count(12)
```

Out[82]:

```
3
```

In [83]:

```
A.count(9999)
```

Out[83]:

```
0
```

In [84]:

```
A = [1, 12, 12, [4, 5], [7, 7], [9, 9, 10]]
print(A)
```

```
[1, 12, 12, [4, 5], [7, 7], [9, 9, 10]]
```

In [86]:

```
A.count([2,7])
```

Out[86]:

```
0
```

In [87]:

```
A.count([7,7])
```

Out[87]:

```
1
```

In [90]:

```
A = [1, 12, 12, 12, 5, 6, 7]
for i in A:
    print(i,A.count(i))
```

```
1 1
12 3
12 3
12 3
5 1
6 1
7 1
```

In [91]:

```
A = "JYOTI"
for i in A:
    print(i)
```

```
J
Y
O
T
I
```

In [92]:

```
A = "ANCHAL"
for i in A:
    print(i , A.count(i))
```

```
A 2
N 1
C 1
H 1
A 2
L 1
```

In [94]:

```
A = ["ANCHAL" , "JYOTI", "ANCHAL", "SUNNY", "ANCHAL"]
for i in A:
    print(i, A.count(i))
```

```
ANCHAL 3
JYOTI 1
ANCHAL 3
SUNNY 1
ANCHAL 3
```

EXTEND

In [95]:

```
a = [1, 2, 3, 4]
b = [2, 4, 6, 8]

a+b
```

Out[95]:

```
[1, 2, 3, 4, 2, 4, 6, 8]
```

In [97]:

```
a = [1, 2, 3, 4]
b = [2, 4, 6, 8]
a.append(b)

a
```

Out[97]:

```
[1, 2, 3, 4, [2, 4, 6, 8]]
```

In [98]:

```
a = [1, 2, 3, 4]
b = [2, 4, 6, 8]
a.extend(b)

a
```

Out[98]:

```
[1, 2, 3, 4, 2, 4, 6, 8]
```

In [ ]: