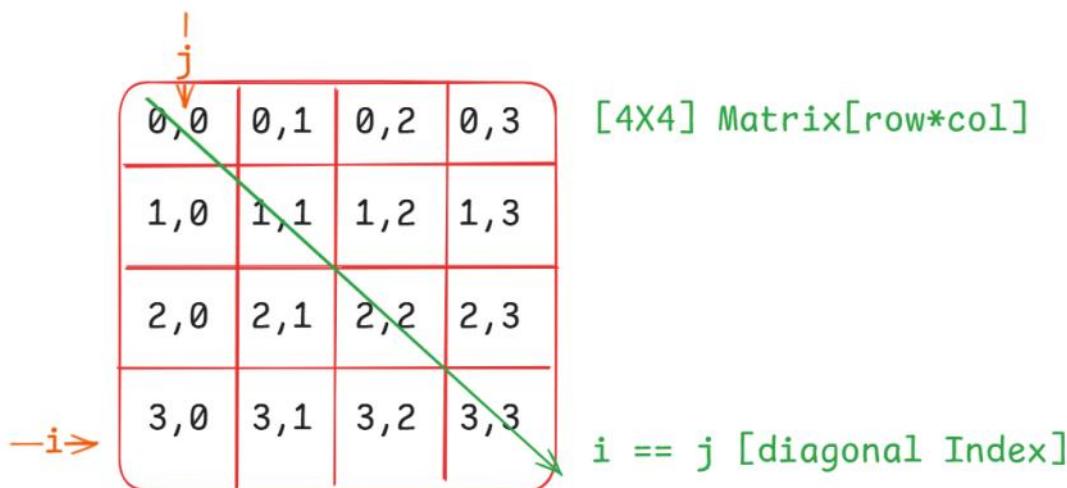


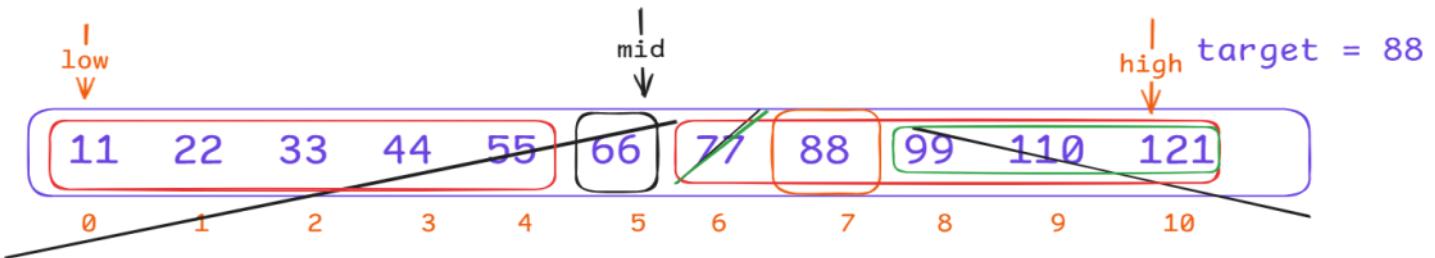
Looping Statement - III & Control Transfer Statement

Session Objectives

- Understand is nested for/while loop
- Understand what a while loop is
- What control transfer statements are
- How to use the break statement
- How to use the continue statement
- How to use the pass statement



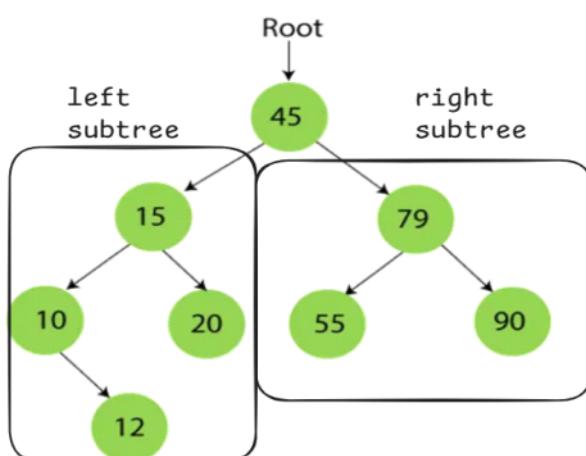
```
car_list = [  
    ['Taigun', 'Creta', 'Slavia', 'Venue'],  
    ['Sierra', 'City', 'Curv', 'Harrier'],  
    ['Safari', 'Lord Alto', 'Thar', 'Virtus'],  
    ['Defender', 'Innova', 'Baleno', 'Legender'],  
    ['ScorpioN', 'Grand Vitara', 'Brezza', 'XUV700']]  
  
reverse_car_list = []  
outer_stop = len(car_list) # 5  
for i in range(outer_stop-1, -1, -1): # [4,3,2,1,0]  
    inner_stop = len(car_list[0]) # 4  
    for j in range(inner_stop-1, -1, -1): [3,2,1,0]  
        reverse_car_list.append(car_list[i][j])  
  
reverse_car_list
```



```
mid = (low+high)//2
```

$$(0 + 10)/2 = 5$$

```
if arr[mid] == target:  
    print(mid)  
elif target > arr[mid]:  
    low = mid + 1  
elif target < arr[mid]:  
    high = mid - 1
```



$$mid = (6+10) // 2 == 16//2 = 8$$

$$mid = (6+7)//2 == 6$$

$$mid = (7+7) //2 == 7$$

```
num_list = [11,22,33,44,55,66,77,88,99,110,121]  
# Binary Search  
low = 0  
high = len(num_list) - 1  
target = 99  
while low <= high:  
    mid = (low+high)//2  
    if num_list[mid] == target:  
        print(f"Value is found at {mid} index.")  
        break  
    elif target > num_list[mid]:  
        low = mid+1  
    elif target < num_list[mid]:  
        high = mid-1
```

```
# 5 X 5 Matrix[row][col]  
i = 0  
while i < 5: True [0,1,2,3,4]  
    j = 0  
    while j < 5: True [0,1,2,3,4]  
        print(f"{i,j}" , end = " ")  
        j+=1  
    print()  
    i+=1  
print("5*5 Matrix Print Successfully ✓")  
(0, 0) (0, 1) (0, 2) (0, 3) (0, 4)  
(1, 0) (1, 1) (1, 2) (1, 3) (1, 4)  
(2, 0) (2, 1) (2, 2) (2, 3) (2, 4)  
(3, 0) (3, 1) (3, 2) (3, 3) (3, 4)  
(4, 0) (4, 1) (4, 2) (4, 3) (4, 4)  
5*5 Matrix Print Successfully ✓
```

Memory

```
i = 0-1-2-3-4  
j = 0-1-2-3-4  
  
i = 0  
    j = 0,1,2,3,4  
i = 1  
    j = 0,1,2,3,4
```

```

# Generator Expression -> () -> instead [] [List Comprehension]
cube_generator = (x**3 for x in range(1,11)) # [1,2,3,4,5,6,7,8,9,10]
print(cube_generator)
<generator object <genexpr> at 0x000002D6A43EFD30>

tuple(cube_generator)
(1, 8, 27, 64, 125, 216, 343, 512, 729, 1000)

list(cube_generator)
[]

# Generator Expression -> () -> instead [] [List Comprehension]
add10_generator = (x+10 for x in range(1,11)) # [1,2,3,4,5,6,7,8,9,10]
print(add10_generator)
<generator object <genexpr> at 0x000002D6A43EF9F0>

list(add10_generator)
[11, 12, 13, 14, 15, 16, 17, 18, 19, 20]

tuple(add10_generator)
()

# Generator Expression -> ()
key_generator = (x for x in range(1,11)) # [1,2,3,4,5..10] |
value_generator = (x**2 for x in range(1,11)) # [1,4,9,16,25....100]
dict(zip(key_generator,value_generator))

{1: 1, 2: 4, 3: 9, 4: 16, 5: 25, 6: 36, 7: 49, 8: 64, 9: 81, 10: 100}

# For Loop:
for loop:
    # Outer Block Code
    for loop:
        # Inner Block Code

```

```
# Problem Statement : Mutliplication Table -> 11 to 20
for i in range(11,21):
    print(f"Mutliplication Table for Value: {i} -> ")
    for j in range(1,11):
        print(f"{i} X {j} = {i*j}")
    print()
```

and so on

```
Mutliplication Table for Value: 11 ->
11 X 1 = 11
11 X 2 = 22
11 X 3 = 33
11 X 4 = 44
11 X 5 = 55
11 X 6 = 66
11 X 7 = 77
11 X 8 = 88
11 X 9 = 99
11 X 10 = 110
```

```
Mutliplication Table for Value: 12 ->
12 X 1 = 12
12 X 2 = 24
12 X 3 = 36
12 X 4 = 48
12 X 5 = 60
12 X 6 = 72
12 X 7 = 84
12 X 8 = 96
12 X 9 = 108
12 X 10 = 120
```

```
# Table of 11:-> 11, 22, 33, 44.....
# Table of 12:-> 12, 24, 36,.....
# Problem Statement : Mutliplication Table -> 11 to 20
for i in range(11,21):
    print(f"Table of {i}:-> ", end = " ")
    for j in range(1,11):
        if j == 10:
            print(f"{i*j}", end = ".")
        else:
            print(f"{i*j}", end = ", ")
    print()
```

```
Table of 11:-> 11, 22, 33, 44, 55, 66, 77, 88, 99, 110.
Table of 12:-> 12, 24, 36, 48, 60, 72, 84, 96, 108, 120.
Table of 13:-> 13, 26, 39, 52, 65, 78, 91, 104, 117, 130.
Table of 14:-> 14, 28, 42, 56, 70, 84, 98, 112, 126, 140.
Table of 15:-> 15, 30, 45, 60, 75, 90, 105, 120, 135, 150.
Table of 16:-> 16, 32, 48, 64, 80, 96, 112, 128, 144, 160.
Table of 17:-> 17, 34, 51, 68, 85, 102, 119, 136, 153, 170.
Table of 18:-> 18, 36, 54, 72, 90, 108, 126, 144, 162, 180.
Table of 19:-> 19, 38, 57, 76, 95, 114, 133, 152, 171, 190.
Table of 20:-> 20, 40, 60, 80, 100, 120, 140, 160, 180, 200.
```

```
# Index in 2D Array [row][col] [5*4] -> 5 rows & 4 columns
car_list = [
    ['Taigun', 'Creta', 'Slavia', 'Venue'],
    ['Sierra', 'City', 'Curv', 'Harrier'],
    ['Safari', 'Lord Alto', 'Thar', 'Virtus'],
    ['Defender', 'Innova', 'Baleno', 'Legender'],
    ['ScorpioN', 'Grand Vitara', 'Brezza', 'XUV700']
]
for row in car_list:
    print(row)
['Taigun', 'Creta', 'Slavia', 'Venue']
['Sierra', 'City', 'Curv', 'Harrier']
['Safari', 'Lord Alto', 'Thar', 'Virtus']
['Defender', 'Innova', 'Baleno', 'Legender']
['ScorpioN', 'Grand Vitara', 'Brezza', 'XUV700']
```

```
# Index in 2D Array [row][col] [5*4] -> 5 rows & 4 columns
car_list = [
    ['Taigun', 'Creta', 'Slavia', 'Venue'],
    ['Sierra', 'City', 'Curv', 'Harrier'],
    ['Safari', 'Lord Alto', 'Thar', 'Virtus'],
    ['Defender', 'Innova', 'Baleno', 'Legender'],
    ['ScorpioN', 'Grand Vitara', 'Brezza', 'XUV700']
]
# ✗Not Recommended✗
print(car_list[0])
print(car_list[1])
print(car_list[2])
print(car_list[3])
print(car_list[4])
```

['Taigun', 'Creta', 'Slavia', 'Venue']
 ['Sierra', 'City', 'Curv', 'Harrier']
 ['Safari', 'Lord Alto', 'Thar', 'Virtus']
 ['Defender', 'Innova', 'Baleno', 'Legender']
 ['ScorpioN', 'Grand Vitara', 'Brezza', 'XUV700']

```
# Index in 2D Array [row][col] [5*4] -> 5 rows & 4 columns
car_list = [
    ['Taigun', 'Creta', 'Slavia', 'Venue'],
    ['Sierra', 'City', 'Curv', 'Harrier'],
    ['Safari', 'Lord Alto', 'Thar', 'Virtus'],
    ['Defender', 'Innova', 'Baleno', 'Legender'],
    ['ScorpioN', 'Grand Vitara', 'Brezza', 'XUV700']
]
stop = len(car_list) # 5
for i in range(stop): # [0,1,2,3,4]
    print(car_list[i])
```

['Taigun', 'Creta', 'Slavia', 'Venue']
 ['Sierra', 'City', 'Curv', 'Harrier']
 ['Safari', 'Lord Alto', 'Thar', 'Virtus']
 ['Defender', 'Innova', 'Baleno', 'Legender']
 ['ScorpioN', 'Grand Vitara', 'Brezza', 'XUV700']

```
# Index in 2D Array [row][col] [5*4] -> 5 rows & 4 columns
car_list = [
    ['Taigun', 'Creta', 'Slavia', 'Venue'],
    ['Sierra', 'City', 'Curv', 'Harrier'],
    ['Safari', 'Lord Alto', 'Thar', 'Virtus'],
    ['Defender', 'Innova', 'Baleno', 'Legender'],
    ['ScorpioN', 'Grand Vitara', 'Brezza', 'XUV700']
]
for row in car_list:
    for car in row:
        print(car, end = " ")
    print()
```

Taigun Creta Slavia Venue
 Sierra City Curv Harrier
 Safari Lord Alto Thar Virtus
 Defender Innova Baleno Legender
 ScorpioN Grand Vitara Brezza XUV700

```
# Flattening the 2D Array -> 1D Array
car_list = [
    ['Taigun', 'Creta', 'Slavia', 'Venue'],
    ['Sierra', 'City', 'Curv', 'Harrier'],
    ['Safari', 'Lord Alto', 'Thar', 'Virtus'],
    ['Defender', 'Innova', 'Baleno', 'Legender'],
    ['ScorpioN', 'Grand Vitara', 'Brezza', 'XUV700']
]
flatten_car_list = []
for row in car_list:
    for car in row:
        flatten_car_list.append(car)

flatten_car_list
```

```
['Taigun',
 'Creta',
 'Slavia',
 'Venue',
 'Sierra',
 'City',
 'Curv',
 'Harrier',
 'Safari',
 'Lord Alto',
 'Thar',
 'Virtus',
 'Defender',
 'Innova',
 'Baleno',
 'Legender',
 'ScorpioN',
 'Grand Vitara',
 'Brezza',
 'XUV700']
```

```
# Index in 2D Array [row][col] [5*4] -> 5 rows & 4 columns
car_list = [
    ['Taigun', 'Creta', 'Slavia', 'Venue'], # ['Venue', 'Slavia', 'Creta', 'Taigun']
    ['Sierra', 'City', 'Curv', 'Harrier'],
    ['Safari', 'Lord Alto', 'Thar', 'Virtus'],
    ['Defender', 'Innova', 'Baleno', 'Legender'],
    ['ScorpioN', 'Grand Vitara', 'Brezza', 'XUV700']
]
new_car_list = []
for row in car_list:
    reversed_row = row[::-1]
    new_car_list.append(reversed_row)

new_car_list
```

```
[['Venue', 'Slavia', 'Creta', 'Taigun'],
 ['Harrier', 'Curv', 'City', 'Sierra'],
 ['Virtus', 'Thar', 'Lord Alto', 'Safari'],
 ['Legender', 'Baleno', 'Innova', 'Defender'],
 ['XUV700', 'Brezza', 'Grand Vitara', 'ScorpioN']]
```

```
# Flattening the 2D Array -> 1D Array
car_list = [
    ['Taigun', 'Creta', 'Slavia', 'Venue'],
    ['Sierra', 'City', 'Curv', 'Harrier'],
    ['Safari', 'Lord Alto', 'Thar', 'Virtus'],
    ['Defender', 'Innova', 'Baleno', 'Legender'],
    ['ScorpioN', 'Grand Vitara', 'Brezza', 'XUV700']
]
flatten_car_list = []
for row in car_list: # Top To Bottom
    for car in row: # Left to Right
        flatten_car_list.append(car)

print(flatten_car_list[::-1])
```

```
['XUV700', 'Brezza', 'Grand Vitara', 'ScorpioN', 'Legender', 'Baleno', 'Innova', 'Defender', 'Virtus', 'Thar',
 'Lord Alto', 'Safari', 'Harrier', 'Curv', 'City', 'Sierra', 'Venue', 'Slavia', 'Creta', 'Taigun']
```

```
# Flattening the 2D Array -> 1D Array [row][col]
car_list = [
    ['Taigun', 'Creta', 'Slavia', 'Venue'],
    ['Sierra', 'City', 'Curv', 'Harrier'],
    ['Safari', 'Lord Alto', 'Thar', 'Virtus'],
    ['Defender', 'Innova', 'Baleno', 'Legender'],
    ['ScorpioN', 'Grand Vitara', 'Brezza', 'XUV700']
]
car_list[::-1]

[['ScorpioN', 'Grand Vitara', 'Brezza', 'XUV700'],
 ['Defender', 'Innova', 'Baleno', 'Legender'],
 ['Safari', 'Lord Alto', 'Thar', 'Virtus'],
 ['Sierra', 'City', 'Curv', 'Harrier'],
 ['Taigun', 'Creta', 'Slavia', 'Venue']]
```

```
# Flattening the 2D Array -> 1D Array
car_list = [
    ['Taigun', 'Creta', 'Slavia', 'Venue'],
    ['Sierra', 'City', 'Curv', 'Harrier'],
    ['Safari', 'Lord Alto', 'Thar', 'Virtus'],
    ['Defender', 'Innova', 'Baleno', 'Legender'],
    ['ScorpioN', 'Grand Vitara', 'Brezza', 'XUV700']
]
flatten_car_list = []
for row in car_list: # Top To Bottom
    for car in row[::-1]: # Right to Left
        print(car, end = " ")
    print()

Venue Slavia Creta Taigun
Harrier Curv City Sierra
Virtus Thar Lord Alto Safari
Legender Baleno Innova Defender
XUV700 Brezza Grand Vitara ScorpioN
```

```
# Flattening the 2D Array -> 1D Array
car_list = [
    ['Taigun', 'Creta', 'Slavia', 'Venue'],
    ['Sierra', 'City', 'Curv', 'Harrier'],
    ['Safari', 'Lord Alto', 'Thar', 'Virtus'],
    ['Defender', 'Innova', 'Baleno', 'Legender'],
    ['ScorpioN', 'Grand Vitara', 'Brezza', 'XUV700']
]
reverse_car_list = []
for row in car_list[::-1]: # Bottom To Top
    for car in row[::-1]: # Right To Left
        reverse_car_list.append(car)

print(reverse_car_list, end = " ")

[XUV700, 'Brezza', 'Grand Vitara', 'ScorpioN', 'Legender', 'Baleno', 'Innova', 'Defender', 'Virtus', 'Thar',
 'Lord Alto', 'Safari', 'Harrier', 'Curv', 'City', 'Sierra', 'Venue', 'Slavia', 'Creta', 'Taigun']
```

```

# Flattening the 2D Array -> 1D Array
car_list = [
    ['Taigun', 'Creta', 'Slavia', 'Venue'],
    ['Sierra', 'City', 'Curv', 'Harrier'],
    ['Safari', 'Lord Alto', 'Thar', 'Virtus'],
    ['Defender', 'Innova', 'Baleno', 'Legender'],
    ['ScorpioN', 'Grand Vitara', 'Brezza', 'XUV700']
]
reverse_car_list = []
for row in car_list[::-1]: # Bottom To Top
    reverse_car_list.append(row[::-1])

reverse_car_list
[[['XUV700', 'Brezza', 'Grand Vitara', 'ScorpioN'],
 ['Legender', 'Baleno', 'Innova', 'Defender'],
 ['Virtus', 'Thar', 'Lord Alto', 'Safari'],
 ['Harrier', 'Curv', 'City', 'Sierra'],
 ['Venue', 'Slavia', 'Creta', 'Taigun']]]

```

```

# Reversing the elements of 2D array using range
car_list = [
    ['Taigun', 'Creta', 'Slavia', 'Venue'],
    ['Sierra', 'City', 'Curv', 'Harrier'],
    ['Safari', 'Lord Alto', 'Thar', 'Virtus'],
    ['Defender', 'Innova', 'Baleno', 'Legender'],
    ['ScorpioN', 'Grand Vitara', 'Brezza', 'XUV700']
]
outer_stop = len(car_list) # 5
inner_stop = len(car_list[0]) # 4
print(outer_stop)
print(inner_stop)

```

5

4

```

i = -1
j = -1
car_list[i][j] # 'XUV700'

```

'XUV700'

```

# using Range(start,step,stop): # (outer_stop-1,-1,-1)
# using Range(start,step,stop): # (inner_stop-1,-1,-1)
# car_list[row][col]
car_list = [
    ['Taigun','Creta','Slavia','Venue'],
    ['Sierra','City','Curv','Harrier'],
    ['Safari','Lord Alto','Thar','Virtus'],
    ['Defender','Innova','Baleno','Legender'],
    ['ScorpioN','Grand Vitara','Brezza','XUV700']
]
reverse_car_list = []
outer_stop = len(car_list) # 5
for i in range(outer_stop-1,-1,-1): #
    inner_stop = len(car_list[0]) # 4
    for j in range(inner_stop-1,-1,-1):
        reverse_car_list.append(car_list[i][j])

reverse_car_list

```

```

['XUV700',
'Brezza',
'Grand Vitara',
'ScorpioN',
'Legender',
'Baleno',
'Innova',
'Defender',
'Virtus',
'Thar',
'Lord Alto',
'Safari',
'Harrier',
'Curv',
'City',
'Sierra',
'Venue',
'Slavia',
'Creta',
'Taigun']

```

```

# using Range(start,step,stop): # (outer_stop-1,-1,-1)
# using Range(start,step,stop): # (inner_stop-1,-1,-1)
# car_list[row][col]
car_list = [
    ['Taigun','Creta','Slavia','Venue'],
    ['Sierra','City','Curv','Harrier'],
    ['Safari','Lord Alto','Thar','Virtus'],
    ['Defender','Innova','Baleno','Legender'],
    ['ScorpioN','Grand Vitara','Brezza','XUV700']
]
reverse_row_list = []
outer_stop = len(car_list) # 5
for i in range(outer_stop): # [0,1,2,3,4]
    inner_stop = len(car_list[0]) # 4
    for j in range(inner_stop-1,-1,-1): # [3,2,1,0]
        reverse_row_list.append(car_list[i][j])

reverse_row_list

```

```

['Venue',
'Slavia',
'Creta',
'Taigun',
'Harrier',
'Curv',
'City',
'Sierra',
'Virtus',
'Thar',
'Lord Alto',
'Safari',
'Legender',
'Baleno',
'Innova',
'Defender',
'XUV700',
'Brezza',
'Grand Vitara',
'ScorpioN']

```

```

# While Loop:
#   # Initializer
#   While Condition: # Test Condition [Boolean]
#     # Code Block
#     # incremental / decremental

val = 10
while val>=1:
    print(f"{val} : Hello World! ")
    val= val- 1

10 : Hello World!
9 : Hello World!
8 : Hello World!
7 : Hello World!
6 : Hello World!
5 : Hello World!
4 : Hello World!
3 : Hello World!
2 : Hello World!
1 : Hello World!

```

```

# Infinite Loop
val = 10
while val>=1:
    print(f"{val} : Hello World! ")
    val= val + 1

car_list = ['Taigun','Creta','Slavia','Venue','Sierra','City','Curv',
           'Harrier','Safari','Lord Alto','Thar','Virtus','Defender',
           'Innova','Baleno','Legender','ScorpioN','Grand Vitara']
len(car_list) # 18

18

idx = 0 # Start
while idx < len(car_list): # Stop
    print(car_list[idx], end = " ")
    idx+=1 # Step

Taigun Creta Slavia Venue Sierra City Curv Harrier Safari Lord Alto Thar Virtus Defender Innova Baleno Legender ScorpioN Grand Vitara

```

```

idx = 0 # Start
while idx < len(car_list): # Stop
    print(car_list[idx], end = " ")
    idx+=2 # Step

Taigun Slavia Sierra Curv Safari Thar Defender Baleno ScorpioN

idx = 1 # Start
while idx <= len(car_list): # Stop
    print(car_list[idx-1], end = " ")
    idx+=1 # Step

Taigun Creta Slavia Venue Sierra City Curv Harrier Safari Lord Alto Thar Virtus Defender Innova Baleno Legender ScorpioN Grand Vitara

# Reverse Iteration
idx = len(car_list) - 1 # Start [17]
while idx >= 0: # Stop
    print(car_list[idx], end = " ")
    idx-=1 # Step

Grand Vitara ScorpioN Legender Baleno Innova Defender Virtus Thar Lord Alto Safari Harrier Curv City Sierra Venue Slavia Creta Taigun

```

```

for i in range(len(car_list)): # (start=0,stop,step=1)
    print(car_list[i] , end = " ")
Taigun Creta Slavia Venue Sierra City Curv Harrier Safari Lord Alto Thar Virtus Defender Innova Baleno Legende
r ScorpioN Grand Vitara

num_list = [11,22,33,44,55,66,77,88,99,110,121]
# Binary Search
low = 0
high = len(num_list) - 1
target = 99
while low <= high:
    mid = (low+high)//2
    if num_list[mid] == target:
        print(f"Value is found at {mid} index.")
        break
    elif target > num_list[mid]:
        low = mid+1
    elif target < num_list[mid]:
        high = mid-1
else: # Only run when while Loop successfully works.
    print(f"{target} is not found in our num_list.")

```

Value is found at 8 index.

```

num_list = [11,22,33,44,55,66,77,88,99,110,121]
# Binary Search
low = 0
high = len(num_list) - 1
target = 77
while low <= high:
    mid = (low+high)//2
    if num_list[mid] == target:
        print(f"Value is found at {mid} index.")
        break
    elif target > num_list[mid]:
        low = mid+1
    elif target < num_list[mid]:
        high = mid-1
else: # Only run when while Loop successfully works.
    print(f"{target} is not found in our num_list.")

```

Value is found at 6 index.

```

num_list = [11,22,33,44,55,66,77,88,99,110,121]
# Binary Search
low = 0
high = len(num_list) - 1
target = 29
while low <= high:
    mid = (low+high)//2
    if num_list[mid] == target:
        print(f"Value is found at {mid} index.")
        break
    elif target > num_list[mid]:
        low = mid+1
    elif target < num_list[mid]:
        high = mid-1
else: # Only run when while Loop successfully works.
    print(f"{target} is not found in our num_list.")

29 is not found in our num_list.

```

```

# Range -> If-Else in While Loop
val = int(input("Enter the Value: "))
counter = 0
while counter <= val:
    if counter % 2 == 0: # Even Case
        print(f"{counter} is an even value")
    else: # Odd Case
        print(f"{counter} is an odd value")
    counter+=1

```

Enter the Value: 5
0 is an even value
1 is an odd value
2 is an even value
3 is an odd value
4 is an even value
5 is an odd value

```

# Range -> If-Else in While Loop
val = int(input("Enter the Value: "))
counter = 0
while counter <= val:
    if counter % 2 == 0: # Even Case
        print(f"{counter} is an even value")
        counter+=1
    else: # Odd Case
        print(f"{counter} is an odd value")
        counter+=1

```

Enter the Value: 7
0 is an even value
1 is an odd value
2 is an even value
3 is an odd value
4 is an even value
5 is an odd value
6 is an even value
7 is an odd value

```

# Range -> If-Else in While Loop
# Infinite Loop
val = int(input("Enter the Value: "))
counter = 0
while counter <= val:
    if counter % 2 == 0: # Even Case
        print(f"{counter} is an even value")
    else: # Odd Case
        print(f"{counter} is an odd value")
counter+=1

```

```

# While : Else
val = 10 # Start
while val >=1: # stop
    print(val , end = " ")
    if val == 7:
        break # Abruptly breaking the Loop
    val = val - 1 # step
else:
    print("Loop Run Successfully ✅")

```

10 9 8 7

```

# While : Else
val = 10 # Start
while val >=1: # stop
    if val == 7:
        break # Abruptly breaking the Loop
    print(val , end = " ")
    val = val - 1 # step
else:
    print("Loop Run Successfully ✅")

```

10 9 8

```

# While : Else
val = 10 # Start
while val >=1: # stop
    if val == 0:
        break # Abruptly breaking the Loop
    print(val , end = " ")
    val = val - 1 # step
else:
    print("Loop Run Successfully ✅")

```

10 9 8 7 6 5 4 3 2 1 Loop Run Successfully ✅

```

# While : Else
val = 10 # Start
while val >=1: # stop
    if val == 0:
        break # Abruptly breaking the Loop
    print(val , end = " ")
    val = val - 1 # step
else:
    print()
    print("Loop Run Successfully ✅")

```

10 9 8 7 6 5 4 3 2 1
Loop Run Successfully ✅

```

# Shorthand while Loop
val = 1
while val<=11 : print(val , end = " "); val+=2

```

1 3 5 7 9 11

```

# Shorthand while Loop
val = 0
while val<=10 : print(val , end = " "); val+=2

```

0 2 4 6 8 10

```

# Nested While Loop:
While Outer Condition:
    # Outer Loop Body
    While Inner Condition:
        # Inner Loop Body

```

```

# 5 X 5 Matrix[row][col]
i = 0
while i < 5:
    j = 0
    while j < 5:
        print(f"{i,j}" , end = " ")
        j+=1
    print()
    i+=1
print("5*5 Matrix Print Successfully ✅")

```

(0, 0) (0, 1) (0, 2) (0, 3) (0, 4)
(1, 0) (1, 1) (1, 2) (1, 3) (1, 4)
(2, 0) (2, 1) (2, 2) (2, 3) (2, 4)
(3, 0) (3, 1) (3, 2) (3, 3) (3, 4)
(4, 0) (4, 1) (4, 2) (4, 3) (4, 4)

5*5 Matrix Print Successfully ✅

```

# 5 X 5 Matrix[row][col]
i = 0
while i < 5:
    j = 0
    while j < 5:
        print(f"{i,j} -> {i*j}" , end = " ")
        j+=1
    print()
    i+=1
print("5*5 Matrix Print Successfully ✅")

```

(0, 0) -> 0 (0, 1) -> 0 (0, 2) -> 0 (0, 3) -> 0 (0, 4) -> 0
(1, 0) -> 0 (1, 1) -> 1 (1, 2) -> 2 (1, 3) -> 3 (1, 4) -> 4
(2, 0) -> 0 (2, 1) -> 2 (2, 2) -> 4 (2, 3) -> 6 (2, 4) -> 8
(3, 0) -> 0 (3, 1) -> 3 (3, 2) -> 6 (3, 3) -> 9 (3, 4) -> 12
(4, 0) -> 0 (4, 1) -> 4 (4, 2) -> 8 (4, 3) -> 12 (4, 4) -> 16
5*5 Matrix Print Successfully ✅

```

# 5 X 5 Matrix[row][col]
i = 0
while i < 5:
    j = 0
    while j < 5:
        if i == j : # Diagonal Matrix
            print(f"{i,j}" , end = " ")
        j+=1
    print()
    i+=1
print("5*5 Matrix Print Successfully ✅")

```

(0, 0)
(1, 1)
(2, 2)
(3, 3)
(4, 4)
5*5 Matrix Print Successfully ✅