cars = ["Ford", "Volvo", "BMW"]

for x in cars:

    print("hello world")

cars = ["Ford", "Volvo", "BMW"]

for x in cars:

    print("hello world")

cars = ["Ford", "Volvo", "BMW"]

cars.remove("Volvo")

print(cars)

fruits = ["apple", "banana", "cherry"]

x = fruits.count("cherry")

print(x)

fruits = [1, 4, 2, 9, 7, 8, 9, 3, 1]

x = fruits.count(9)

print(x)

fruits = ["apple", "banana", "cherry"]

cars = ["Ford", "Volvo", "BMW"]

fruits.extend(cars)

print(fruits)

fruits = ["apple", "banana", "cherry"]

x=fruits.index("cherry")

fruits=[5, 55, 64, 32, 16, 32]

x=fruits.index(32)

print(x)

cars = ["Ford", "Volvo", "BMW"]

cars.sort()

print(cars)

cars = [5, 2, 89, 56, 44]

cars.sort()

print(cars)

fruits = ["apple", "banana", "cherry"]

fruits.reverse()

print(fruits)

str1=["Hello World"]

str2=["I love python"]

str3=["Hello World"]

#compare

print(str1==str2)

print(str1==str3)

str1=["Hello World"]

str2=["I love python"]

str3=["Hello World"]

mm = str1 + str2

print(mm)

str1= "hello world"

str2="I LOVE PYTHON"

print(str1.upper())

print(str2.lower())

t=[[1,2,3,4],[1,2,3],[10,4,6,7],[4,3,7,8]]

for r in t:

    for c in r:

        print(c,end =" ")

    print()

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

1. **Add 2 matrix:**

t=[[1,2,3,],

   [11,12,13],

   [10,4,6],

   [4,3,7]]

tt=[[5, 8, 3],

    [33, 23, 6],

    [12, 3, 8],

    [35, 54, 66]]

result = [[0,0,0],

          [0,0,0],

          [0,0,0],

          [0,0,0]]

for i in range (len(t)):

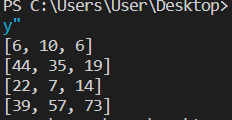
    for j in range(len(t[0])):

            result[i][j]= t[i][j] + tt[i][j]

for r in result:

    print(r)

**Result:**

****

1. **Transpose a Matrix**

t=[[1,2,3,],

   [11,12,13],

   [10,4,6],

   [4,3,7]]

result = [[0,0,0,0],

          [0,0,0,0],

          [0,0,0,0]]

for i in range (len(t)):

    for j in range(len(t[0])):

            result[j][i]= t[i][j]

for r in result:

    print(result)

**Output:**

**A screenshot of a computer program

Description automatically generated with low confidence**

1. **Multiply 2 matrices**

t=[[1,2,3,],

   [11,12,13],

   [10,4,6],

   [4,3,7]] #4\*3 matrix

tt=[[5, 8, 3],

    [33, 23, 6],

    [12, 3, 8]] #3\*3

result = [[0,0,0],

          [0,0,0],

          [0,0,0],

          [0,0,0]] #4\*3

for i in range (len(t)):

    for j in range(len(tt[0])):

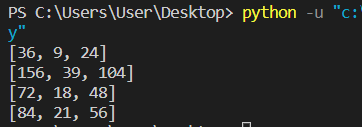
            for k in range(len(tt)):

                result[i][j]= t[i][k] \* tt[k][j]

for r in result:

       print(r)

**Output:**

****

1. **Search an element with linear search**

def linear\_Search(list, position, find):

    for i in range(0, position):

        if (list[i] == find):

            return i

    return -1

list = [5, 9, 56, 32, 55, 21, 6, 90]

find = 89

position = len(list)

result = linear\_Search(list, position, find)

if (result == -1):

    print("Element not found")

else:

    print("Element found at index: ", result)

**Output:**

**A black screen with white text

Description automatically generated with low confidence**

1. **Search an element with binary search**

def binary\_search(array, findArray, lower, higher):

    while lower <= higher:

        mid = lower + (higher - lower)//2

        if array[mid] == findArray:

            return mid

        elif array[mid] < findArray:

            lower = mid + 1

        else:

            higher = mid - 1

    return -1

# we need sorted array

array = [1, 2, 3, 4, 5, 6, 7]

findArray = 6

print("The given array is", array)

index = binary\_search(array, findArray, 0, len(array)-1)

if index != -1:

    print("Element 6 found in index " + str(index))

else:

    print("Element Not found")

**Output:**

****