

```
In [1]: print("\n\n nsti \t\t jodhpur \n")
```

nsti jodhpur

```
In [6]: a=7
print(a)
b=7.0
print(b)
a="nsti jodhpur"
print(a)
```

7
7.0
nsti jodhpur

```
In [7]: num = [1,2,3,4]
print(num)
stringVal= ["name","class","percent"]
print(stringVal)
name= ["kapil","ram","shyam"]
print(name)
print(name[1])
```

[1, 2, 3, 4]
['name', 'class', 'percent']
['kapil', 'ram', 'shyam']
ram

```
In [11]: print("1. Addition");
print("2. Subtraction");
print("3. Multiplication");
print("4. Division");
print("5. Exit");
choice = int(input("enter your Choice 1 2 3 4 5"))
if choice >=1 and choice<=4:
    print("enter two number")
    num1 = int(input("enter first number = "))
    num2 = int(input("enter second number = "))
    if choice ==1:
        res=num1+num2
        print(res)
    elif choice ==2:
        res=num1-num2
    elif choice ==2:
        res=num1*num2
        print(res)
    elif choice ==2:
        res=num1/num2
        print(res)

    else:
        print("enter a valid input")
```

```

1. Addition
2. Subtraction
3. Multiplication
4. Division
5. Exit
enter your Choice 1 2 3 4 51
enter two number
enter first number10
enter second number20
30

```

```

In [15]: x= "welcome"
        y = "nsti"
        print((x+y)*5)

welcomenstiwelcomenstiwelcomenstiwelcomenstiwelcomensti

```

```

In [17]: list = [10,20,545,1,450,546,1,20]
        a = min(list)
        b = max(list)
        print(a, b)

1 546

```

```

In [20]: num = int(input("enter your number = "))
        if num %2 ==0:
            print("this is even number")
        else:
            print("this is odd number")

enter your number = 5
this is odd number

```

```

In [22]: # Function to create a histogram from a List of integers
def create_histogram(numbers):
    histogram = {}

    # Iterate through the List using a for loop
    for num in numbers:
        # Check if the number is already in the histogram
        if num in histogram:
            histogram[num] += 1
        else:
            histogram[num] = 1

    return histogram

# Input List of integers
numbers = [1, 2, 2, 3, 3, 3, 4, 4, 5, 5, 5, 5]

# Create a histogram
histogram = create_histogram(numbers)

# Display the histogram
for num, count in histogram.items():
    print(f"{num}: {'*' * count}")

```

```

1: *
2: **
3: ***
4: **
5: ****

```

```
In [34]: #gcd doing mcf
def gcdfound(a, b):
    if b == 0:
        return abs(a)
    else:
        return gcdfound(b, a % b)

def lcmfound(a,b):
    return (a // gcdfound(a,b))* b

resultgcd = gcdfound(12, 17)
resultlcm = lcmfound (15, 17)

print("The gcd of 12 and 17 is:", resultgcd)
print("The gcd of 15 and 17 is:", resultlcm)
```

The gcd of 12 and 17 is: 1
The gcd of 15 and 17 is: 255

```
In [44]: a = {1: 2, 5: 6, 1: 3, 2: 6, 8: 9, 9: 8}

# Sort the dictionary by values in ascending order
ascending_sorted = dict(sorted(a.items(), key=lambda item: item[1]))

# Sort the dictionary by values in descending order
descending_sorted = dict(sorted(a.items(), key=lambda item: item[1], reverse=True))

print("Original Dictionary:")
print(a)

print("\nSorted by Values (Ascending):")
print(ascending_sorted)

print("\nSorted by Values (Descending):")
print(descending_sorted)
```

Original Dictionary:
{1: 3, 5: 6, 2: 6, 8: 9, 9: 8}

Sorted by Values (Ascending):
{1: 3, 5: 6, 2: 6, 9: 8, 8: 9}

Sorted by Values (Descending):
{8: 9, 9: 8, 5: 6, 2: 6, 1: 3}

```
In [47]: a = (1,2,3,4)
print(type(a))

<class 'tuple'>
```

```
In [51]: #Write a Python program to create a tuple with different data types
a = (True , "nsti",1 ,2, 1.2)
print(a)
print(type(a))

(True, 'nsti', 1, 2, 1.2)
<class 'tuple'>
```

```
In [57]: #Write a Python program to create a set
a = set([1,2,3,4,5,6])
print(a)
print(type(a))
```

```
{1, 2, 34, 5, 6}
<class 'set'>
```

```
In [64]: a = set()
a.add("red") # here we are add only one item
a.update(["blue", "orange", "nsti"])
print(a)

{'red', 'orange', 'nsti', 'blue'}
```

```
In [71]: #Write a Python program to find maximum and the minimum value in a set.
a = set([10,20,320,1000,1,0,15,5])
print(min(a))
print(max(a))

0
1000
```

```
In [74]: #Write a Python program to find the Length of a set (1 Hrs)
a=set([10,20,30,40,50,True])
print(len(a))

6
```

```
In [76]: # f= (c*9/5)+32
# c= (f-32)*5/9

# forin at to cealicus
c = int(input("enter your data = "))
f = (c*9/5)+32
print(f)
# cealicus to forin at

cal = (c-32)*5/9
print(cal)

enter your data = 50
122.0
10.0
```

```
In [84]: # Fibonacci series
# fn = fn-1 + fn-2 and f0 = 0 f1 = 1

def fibo(n):
    if n < 0:
        print("enter a valid input")
    elif n == 0:
        return 0
    elif n == 1 or n == 2:
        return 1
    else:
        data = fibo(n-1) + fibo(n-2)
        return data

n = 9
res = fibo(n)

print(f"The Fibonacci number at index {n} is: {res}")
```

The Fibonacci number at index 9 is: 34

```
In [91]: # factorial number
n=5
fact=1
```

```
for i in range(1, n+1):
    fact = fact * i
    print(fact)
```

```
1
2
6
24
120
```

In []: *#Write a python class to reverse a string word by word*

```
def reversersting(in_word):
    w= in_word.split(" ")
    nw = [i[::-1] for i in w]
    ns = " ".join(nw)
    return ns
in_word = input("enter the string: ")
print(reversersting(in_word))
```

In []: **import** math

```
class Circle:
    def __init__(self, radius):
        self.radius = radius

    def area(self):
        """Compute the area of the circle."""
        return math.pi * self.radius ** 2

    def perimeter(self):
        """Compute the perimeter (circumference) of the circle."""
        return 2 * math.pi * self.radius

# Create an instance of the Circle class with a radius of 5
circle_instance = Circle(5)

# Calculate and print the area and perimeter of the circle
print(f"Radius: {circle_instance.radius}")
print(f"Area: {circle_instance.area()}")
print(f"Perimeter: {circle_instance.perimeter()}")
```

In []: **def** bubble_sort(arr):
n = len(arr)

```
# Traverse through all elements in the list
for i in range(n):
    # Flag to optimize the algorithm by checking if any swaps occurred in this
    swapped = False

    # Last i elements are already in place, so we don't need to check them
    for j in range(0, n - i - 1):
        # Swap if the element found is greater than the next element
        if arr[j] > arr[j + 1]:
            arr[j], arr[j + 1] = arr[j + 1], arr[j] # Swap the elements
            swapped = True

    # If no two elements were swapped in this pass, the list is already sorted
    if not swapped:
        break

# Example usage:
my_list = [64, 25, 12, 22, 11]
```

```
bubble_sort(my_list)

print("Sorted list using bubble sort:")
print(my_list)
```

```
In [ ]: # Define the source and destination file paths
source_file_path = "C:\Users\ibmjo\Downloads\input.txt"
destination_file_path = "C:\Users\ibmjo\Downloads\output.txt"

try:
    # Open the source file in read mode
    with open(source_file_path, 'r') as source_file:
        # Read the contents of the source file
        file_contents = source_file.read()

    # Open the destination file in write mode
    with open(destination_file_path, 'w') as destination_file:
        # Write the contents to the destination file
        destination_file.write(file_contents)

    print(f"Contents of '{source_file_path}' copied to '{destination_file_path}' successfully")

except FileNotFoundError:
    print(f"File '{source_file_path}' not found.")
except Exception as e:
    print(f"An error occurred: {e}")
```

```
In [ ]: #Write a python program to find the frequency of words in a file
#import counter for read data
from collections import Counter
def word_count(fname):
    #Open a file and read text
    with open(fname) as f:
        #Separating a text word by word using split()
        return Counter(f.read().split())
print("Number of words in the file :",word_count("output.txt"))
```

```
In [ ]: # Write a python program to illustrate exception handling
import sys
list=['boy', 'cat',0 ,14.3]
for entry in list:

    try:
        print("the entry is:" , entry)
        r=1/int(entry)
    except(ValueError):
        print("Hey a ValueError exception occurred")
    except(ZeroDivisionError):
        print("Hey a ZeroDivisionError exception occurred")
    except:
        print("some error occur")
        print("the recipocal of the entry is ",r)
```

```
In [ ]: pip install mysql-connector-python
```

```
In [ ]: pip list
```

```
In [ ]: import mysql.connector as c
```

```
In [ ]:
```

