

**PYTHON**

**PROGRAMS**

**SHEKAR C**

**18CS601025**

**V MCA**

**Programs**

1. Write a Python Program to Print all Prime Numbers in an Interval
2. Write a Python Program to Convert Decimal to Binary, Octal and Hexadecimal
3. Write a Python program to get the smallest number from a list.
4. Write a Python program to remove duplicates from a list.
5. Write a Python script to concatenate following dictionaries to create a new one’

Sample Dictionary :  
dic1={1:10, 2:20} dic2={3:30, 4:40} dic3={5:50,6:60}  
Expected Result : {1: 10, 2: 20, 3: 30, 4: 40, 5: 50, 6: 60}

1. Write a Python program to find the highest 3 values in a dictionary.
2. Write a Python program to sort a tuple by its float element.    
   Sample data: [('item1', '12.20'), ('item2', '15.10'), ('item3', '24.5')]  
   Expected Output: [('item3', '24.5'), ('item2', '15.10'), ('item1', '12.20')]
3. Write a Python program to check a given set has no elements in common with other given set.
4. Write a Python program to extract characters from various text files and puts them into a list.
5. Write a NumPy program to create a structured array from given student name, height, class and their data types. Now sort by class, then height if class are equal.

Expected Output:

Original array:

[(b'James', 5, 48.5 ) (b'Nail', 6, 52.5 ) (b'Paul', 5, 42.1 ) (b'Pit', 5, 40.11)]

Sort by age, then height if class are equal:

[(b'Pit', 5, 40.11) (b'Paul', 5, 42.1 ) (b'James', 5, 48.5 ) (b'Nail', 6, 52.5 )]

1. **Write a Python Program to Print all Prime Numbers in an Interval**

**Program:**

**start=int(input("Enter start range :"))**

**end=int(int(input("Enter end range : ")))**

**print(start)**

**print(end)**

**for num in range(start,end+1):**

**if(num>1):**

**for i in range(2, num):**

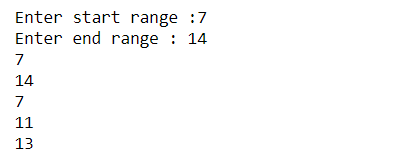
**if (num % i) == 0:**

**break**

**else:**

**print(num)**

**Output:**



1. **Write a Python Program to Convert Decimal to Binary, Octal and Hexadecimal**

Program:

def decimal\_to\_binary(dec):

decimal = int(dec)

# Prints equivalent decimal

print(decimal, " in Binary : ", bin(decimal))

# Function to convert decimal to octal

def decimal\_to\_octal(dec):

decimal = int(dec)

# Prints equivalent decimal

print(decimal, "in Octal : ", oct(decimal))

# Function to convert decimal to hexadecimal

def decimal\_to\_hexadecimal(dec):

decimal = int(dec)

# Prints equivalent decimal

print(decimal, " in Hexadecimal : ", hex(decimal))

# Driver program

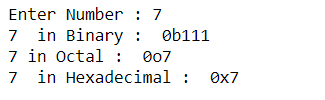
dec = int(input("Enter Number : "))

decimal\_to\_binary(dec)

decimal\_to\_octal(dec)

decimal\_to\_hexadecimal(dec)

Output:



**3.** **Write a Python program to get the smallest number from a list.**

Program:

# Python program to find smallest

# number in a list

input\_string = input("Enter a list elements separated by space ")

print("\n")

l = input\_string.split()

print("The list is ", l)

# Assign first element as a minimum.

min1 = l[0]

for i in range(len(l)):

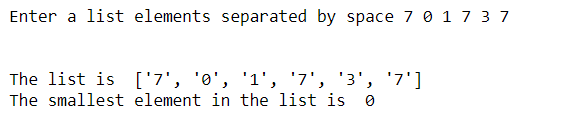
# If the other element is min than first element

if l[i] < min1:

min1 = l[i] # It will change

print("The smallest element in the list is ", min1)

Output:



**4. Write a Python program to remove duplicates from a list.**

Program:

def Remove(duplicate):

final\_list = []

for num in duplicate:

if num not in final\_list:

final\_list.append(num)

return final\_list

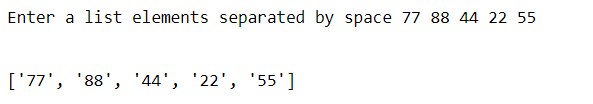
input\_string = input("Enter a list elements separated by space ")

print("\n")

duplicate = input\_string.split()

print(Remove(duplicate))

Output:



**5.Write a Python script to concatenate following dictionaries to create a new one’**

Program:

dic1={1:10, 2:20}

dic2={3:30, 4:40}

dic3={5:50,6:60}

dic4 = {}

for d in (dic1, dic2, dic3): dic4.update(d)

print(dic4)

Output:



**6.Write a Python program to find the highest 3 values in a dictionary.**

Program:

from heapq import nlargest

# Initial Dictionary

my\_dict = {'A': 67, 'B': 23, 'C': 45,

'D': 56, 'E': 12, 'F': 69}

print("Initial Dictionary:")

print(my\_dict, "\n")

ThreeHighest = nlargest(3, my\_dict, key=my\_dict.get)

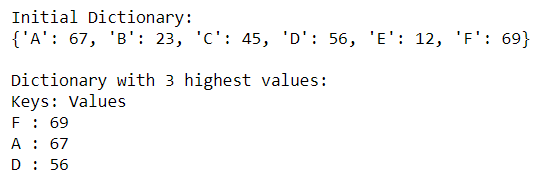
print("Dictionary with 3 highest values:")

print("Keys: Values")

for val in ThreeHighest:

print(val, ":", my\_dict.get(val))

Output:



**7.Write a Python program to sort a tuple by its float element.**

Sample data: [('item1', '12.20'), ('item2', '15.10'), ('item3', '24.5')]  
Expected Output: [('item3', '24.5'), ('item2', '15.10'), ('item1', '12.20')]

Program:

price = [('item1', '12.20'), ('item2', '15.10'), ('item3', '24.5')]

print( sorted(price, key=lambda x: float(x[1]), reverse=True))

Output:



**8.Write a Python program to check a given set has no elements in common with other given set.**

Program:

sn1 = {1,2,3}

sn2 = {4,5,6}

sn3 = {3}

print("Original sets:")

print(sn1)

print(sn2)

print(sn3)

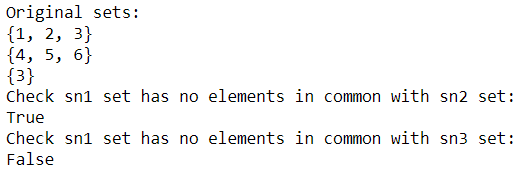
print("Check sn1 set has no elements in common with sn2 set:")

print(sn1.isdisjoint(sn2))

print("Check sn1 set has no elements in common with sn3 set:")

print(sn1.isdisjoint(sn3))

Output:



**9.Write a Python program to extract characters from various text files and puts them into a list.**

Program:

import glob

char\_list = []

files\_list = glob.glob("sample.txt")

for file\_elem in files\_list:

with open(file\_elem, "r") as f:

char\_list.append(f.read())

print(char\_list)

Output:



**10.Write a NumPy program to create a structured array from given student name, height, class and their data types. Now sort by class, then height if class are equal.**

Expected Output:

Original array:

[(b'James', 5, 48.5 ) (b'Nail', 6, 52.5 ) (b'Paul', 5, 42.1 ) (b'Pit', 5, 40.11)]

Sort by age, then height if class are equal:

[(b'Pit', 5, 40.11) (b'Paul', 5, 42.1 ) (b'James', 5, 48.5 ) (b'Nail', 6, 52.5 )]

Program:

import numpy as np

data\_type = [('name', 'S15'), ('class', int), ('height', float)]

students\_details = [('James', 5, 48.5), ('Nail', 6, 52.5),('Paul', 5, 42.10), ('Pit', 5, 40.11)]

# create a structured array

students = np.array(students\_details, dtype=data\_type)

print("Original array:")

print(students)

print("Sort by height")

print(np.sort(students, order='height'))

Output:

