ASSIGNMENT-1

The corrections are added in red.

1. **Python Program to check Armstrong Number.**

Given a number x, determine whether the given number is Armstrong number or not. A positive integer of n digits is called an Armstrong number of order n (order is number of digits) if-

abcd... = pow(a,n) + pow(b,n) + pow(c,n) + pow(d,n) + ....

*Input : 153*

*Output : Yes*

*153 is an Armstrong number.*

*1\*1\*1 + 5\*5\*5 + 3\*3\*3 = 153*

**Sol-**

a=input(“Enter your number:”)  
sum=0  
n=len(a)  
for i in a:  
 i=int(i)  
 sum+=i\*\*n  
if sum==int(a):  
 b="Yes"  
 c="is"  
else:  
 b="no"  
 c="is not"  
print(f"{b}\n{a} {c} an Armstrong number.")

1. **Python Program for Sum of squares of first n natural numbers.**

Given a positive integer N. The task is to find 12+22+32+..+n2.

**Sol-**

n=int(input(“Enter the value of n:”))  
n=n+1  
sum=0  
for i in range(n):  
 sum+=i\*\*2  
print(f"The sum of squares of 1st {n-1} natural numbers is {sum}.")

1. **Python program to find Cumulative sum of a list.**

*Input : list = [10, 20, 30, 40, 50]*

*Output : [10, 30, 60, 100, 150]*

**Sol-**

a=(input("List= ")).strip("[]").split(",")  
ans=[]  
sum=0  
for i in a:  
 sum+=int(i)  
 ans.append(sum)  
print(ans)

def Cumulative(lists):

    cu\_list = []

    length = len(lists)

    cu\_list = [sum(lists[0:x:1]) for x in range(0, length+1)]

    return cu\_list[1:]

1. **Python Program to perform Binary Search.**

**Sol-**

def binary\_search(arr,low,high,x):  
 if low<=high:  
 mid = (low + high) // 2  
 if arr[mid] == x:  
 return mid  
 elif arr[mid] > x:  
 return binary\_search(arr, low, mid - 1, x)  
 else:  
 return binary\_search(arr, mid + 1, high, x)  
 else:  
 return -1  
if \_\_name\_\_=="\_\_main\_\_":  
 arr1 = (input("List= ")).strip("[]").split(",")  
 arr=[]  
 for i in arr1:  
 i = int(i)  
 arr.append(i)  
 x = int(input("which number wanna check= "))  
 result = binary\_search(arr, 0, len(arr) - 1, x)  
 if result != -1:  
 print(f"{x} present in index {result} in the list.")  
 else:  
 print(f"{x} not present in the given list.")

1. **Python Program to perform Bubble Sort.**

**Sol-**

b=input("Enter your list: ").strip("[]").split(",")  
a=[]  
for i in b:  
 a.append(int(i))  
c=0  
for j in range(len(a)-1):  
 c += 1  
 for i in range(len(a) - 1):  
 if a[i] <= a[i + 1]:  
 pass  
 else:  
 try:  
 b = a[i]  
 a[i] = a[i + 1]  
 a[i + 1] = b  
 except:  
 pass  
print(a)

Santosh Kumar Mahapatra