Zed-King

GROUP OF INSTITUTIONS

ISO 9001 : 2015 Certified

M3-R5.1 Practical Marathon



Python Programming

Zed-King Institute

Q1. Write a function to compute the sum of n Terms of the following series. X+X^4/4!+X^6/6!+X^8/8!+…… for any positive integer value of X.

Solution:

```
import math
def series_sum(x, n):
    s = 0
    power = 1    # first term is X^1
    fact = 1

    for i in range(n):
        if i == 0:
            s = s + x
        else:
            power = 2 * i + 2    # 4, 6, 8, ...
            fact = math.factorial(power)
            s = s + (x ** power) / fact

    return s


# Example
x = int(input("Enter value of X: "))
n = int(input("Enter number of terms: "))
print("Sum of series =", series_sum(x, n))
```

OUTPUT:

```
Enter value of X: 2
Enter number of terms: 5
Sum of series = 2.7621869488536155
```

Q2. Write a program that takes as input a list having a mix of 10 positive numbers and a key value.
Apply linear search to find whether the key is present in the list or not.
If input is :[10,50,30,70,80,20,90,40] key=30
Then output should be: yes
If input is :[10,50,30,70,80,20,90,40], key=15
Then output should be: No

Solution:

```
lst = [10, 50, 30, 70, 80, 20, 90, 40]
key = int(input("Enter key: "))

if key in lst:
    print("Yes")
else:
    print("No")
```

output:

```
Enter key:  80
Yes

====================
Enter key:  11
No
```

Learner Institute

Q3. Implement python script to accept line of text and find the number of characters, number of vowels and number of blank spaces in it.

Solution:

```
text = input("Enter a line of text: ")

characters = len(text)
vowels = 0
spaces = 0

for ch in text:
    if ch.lower() in 'aeiou':
        vowels += 1
    elif ch == ' ':
        spaces += 1

print("Number of characters =", characters)
print("Number of vowels =", vowels)
print("Number of blank spaces =", spaces)
```

Output:

```
Enter a line of text: python programming
Number of characters = 18
Number of vowels = 4
Number of blank spaces = 1
```

Q4. Write a python program to display the elements of a list in reverse order.

Solution:

```
lst = []
n = int(input("Enter number of elements: "))

for i in range(n):
    lst.append(int(input("Enter element: ")))

lst.reverse()

print("Reversed list:", lst)
```

Output:

```
Enter number of elements: 5
Enter element: 34
Enter element: 67
Enter element: 36
Enter element: 90
Enter element: 67
Reversed list: [67, 90, 36, 67, 34]
```

Q5. Write python scripts to check string is palindrome or not.

Solution:

```
s = input("Enter a string: ")
rev = ""

for ch in s:
    rev = ch + rev

if s == rev:
    print("Palindrome")
else:
    print("Not Palindrome")
```

Output:

```
Enter a string: raman
Not Palindrome

==========
Enter a string: kanak
Palindrome
```

Q6. Write a python program which returns a string by converting all the character to their opposite letter case (uppercase to lowercase and vice versa).

If input is :Python is Awesome! Then output be: Python is Awesome!

If input is :JoHn CeNa

Then output should be: jOhN cEnA

Solution:

```
s = input("Enter a string: ")
result = ""

for ch in s:
    if ch.isupper():
        result = result + ch.lower()
    elif ch.islower():
        result = result + ch.upper()
    else:
        result = result + ch

print("Output:", result)
```

Output:

```
Enter a string: Python is Awesome!
Output: pYTHON IS aWESOME!

============================== RESTAR
Enter a string: JoHn CeNa

Output: jOhN cEnA
```

Q7. Create a python program that removes even duplicate positive integer number(includes zero) from a list and prints the unique numbers in the order they first appeared.

If input is 4 3 2 1 , then output should be 4 3 2 1.

If input is 0 0 0 1, then output should be 0 1.

If input is 1 1 3 3, then output should be 1 1 3 3.

Solution:

```
lst = []
n = int(input("Enter number of elements: "))

for i in range(n):
    lst.append(int(input("Enter number: ")))

result = []
even_used = []

for num in lst:
    if num % 2 == 0:            # even number (including 0)
        if num not in even_used:
            result.append(num)
            even_used.append(num)
    else:                       # odd number
        result.append(num)

print("Output:", result)
```

Output:

```
Enter number of elements: 4
Enter number: 4
Enter number: 3
Enter number: 2
Enter number: 1
Output: [4, 3, 2, 1]
```

Q8. **Given an integer number n, write a python program to print the factorial of this number.**

If input is :4 then output should be:24.
If input is :5 then output should be:120.
If input is :6 then output should be:720.
If input is :7 then output should be:5040.
If input is :8 then output should be:40320.
If input is :10 then output should be:3608800.

Solution:

```
n = int(input("Enter a number: "))
fact = 1

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

print("Factorial =", fact)
```

Output:

```
Enter a number: 5
Factorial = 120
```

Neg Institute

Q9. Implement python scrip to generate prime numbers series up to n.

Solution:

```
n = int(input("Enter value of n: "))

for num in range(2, n + 1):
    is_prime = True

    for i in range(2, num):
        if num % i == 0:
            is_prime = False
            break

    if is_prime:
        print(num, end=" ")
```

Output:

```
Enter value of n: 50
2 3 5 7 11 13 17 19 23 29 31 37 41 43 47
```

Q10. write a program that takes 2 numbers as command line arguments and prints its sum.

Solution:

```
import sys
a = int(sys.argv[1])
b = int(sys.argv[2])

print("Sum =", a + b)
```

Output:

```
C:\python>110.py 10 20
Sum = 30

C:\python>
```

Q11. Implement python scripts to check given number is Armstrong or not.

Solution:

```
num = int(input("Enter a number: "))
temp = num
sum = 0
digits = len(str(num))

while temp > 0:
    d = temp % 10
    sum = sum + (d ** digits)
    temp = temp // 10

if sum == num:
    print("Armstrong number")
else:
    print("Not an Armstrong number")
```

Output:

```
--------------------------------
Enter a number: 153
Armstrong number

================================
Enter a number: 123
Not an Armstrong number
```

Q12.write a python program to calculate the area of a triangle.

Area=0.5*base*height

Solution:

```
base = float(input("Enter base: "))
height = float(input("Enter height: "))

area = 0.5 * base * height

print("Area of triangle =", area)
```

Output:

```
Enter base: 2
Enter height: 4
Area of triangle = 4.0
```

Q13. Write a python program that:

a) Accept a sentence from the user as input.

b) Display the first 5 character and the last 5 character of the sentence using slicing.

c) Concatenate the sentence with another predefined string.

Solution:

```
sentence = input("Enter a sentence: ")

# First 5 characters
print("First 5 characters:", sentence[:5])

# Last 5 characters
print("Last 5 characters:", sentence[-5:])

# Concatenate with predefined string
new_sentence = sentence + " - Python Programming"
print("After concatenation:", new_sentence)
```

Output:

```
Enter a sentence: zed-king
First 5 characters: zed-k
Last 5 characters: king
After concatenation: zed-king  - Python Programming
```

Q14. write a python program to find the length of a dictionary.iterate through a tuple using for loop.

If input is:{"England":"London","Italy":"Rome"}
then output should be : 2
If input is:{10:"ten",20:"twenty",30:"thirty"}
then output should be : 3
If input is:{}
then output should be : 0

Solution:

```
# Using eval to directly take dictionary input
user_dict = eval(input("Enter a dictionary: "))

# Finding the length of dictionary
dict_length = len(user_dict)

# Printing the length
print("Length of the dictionary is:", dict_length)

# Extra: iterating through dictionary keys as a tuple using a for loop
tuple_keys = tuple(user_dict.keys())
print("Iterating through dictionary keys as tuple:")
for key in tuple_keys:
    print(key)
```

Output:

```
Enter a dictionary: {10:"ten",20:"twenty",30:"thirty"}
Length of the dictionary is: 3
Iterating through dictionary keys as tuple:
10
20
30
```

Zeeshan Institute

Q15.Create a python program that finds the second smallest number in a list of positive integers(including zero).

If input is:5 4 3 2 1 then output should be : 2

If input is:0 0 0 0 1 then output should be : 1

If input is: 1 11 then output should be : 11

Solution:

```
# Input numbers as space-separated string
input_str = input("Enter numbers separated by space: ")

# Convert input to a list of integers without using map()
numbers = []
for num in input_str.split():
    numbers.append(int(num))

# Remove duplicates by converting to a set, then back to a list
unique_numbers = list(set(numbers))

# Sort the unique numbers
unique_numbers.sort()

# Find second smallest number
if len(unique_numbers) >= 2:
    print(unique_numbers[1])
else:
    # If only one unique number, that is the second smallest too
    print(unique_numbers[0])
```

Output:

```
Enter numbers separated by space: 10 55 33 77 3 33 22
10
```

Q16. Write a python program to open the file with "a" for appending,then add a list of texts to append to the file.

Solution:

```
# List of texts to append
texts_to_append = ["Hello World\n", "Python is fun\n", "File handling example\n"]

# Open file in append mode
with open("example.txt", "a") as file:
    for line in texts_to_append:
        file.write(line)

print("Texts have been appended to the file successfully.")
```

Output:

```
Texts have been appended to the file successfully.
```

Q17. Write a python program that takes two integers m and n (where m<n) as input and computes the sum of all even numbers between m and n(inclusive). Use the pass statement for numbers that are not even.

Solution:

```
# Input two integers
m = int(input("Enter the first integer (m): "))
n = int(input("Enter the second integer (n): "))

# Initialize sum
even_sum = 0

# Loop from m to n (inclusive)
for num in range(m, n + 1):
    if num % 2 != 0:
        pass  # Skip odd numbers
    else:
        even_sum += num  # Add even numbers to sum

# Print the result
print("Sum of even numbers between", m, "and", n, "is:", even_sum)
```

Output:

```
Enter the first integer (m): 12
Enter the second integer (n): 45
Sum of even numbers between 12 and 45 is: 476
```

Q18. Write a python program for finding biggest number among 3 numbers.

Solution:

```
# Input three numbers from the user
num1 = float(input("Enter first number: "))
num2 = float(input("Enter second number: "))
num3 = float(input("Enter third number: "))

# Using if-elif-else to find the biggest number
if num1 >= num2 and num1 >= num3:
    biggest = num1
elif num2 >= num1 and num2 >= num3:
    biggest = num2
else:
    biggest = num3

# Print the biggest number
print("The biggest number is:", biggest)
```

Output:

```
Enter first number: 24
Enter second number: 55
Enter third number: 33
The biggest number is: 55.0
```

Q19. Design a Simple calculator using if-elif-else statements.

Solution:

```
# Simple Calculator in Python

# Input two numbers
num1 = float(input("Enter first number: "))
num2 = float(input("Enter second number: "))

# Input the operation
print("Select operation: + for addition, - for subtraction, * for multiplication, / for division")
operation = input("Enter operation: ")

# Using if-elif-else to perform calculation
if operation == '+':
    result = num1 + num2
    print("Result:", result)
elif operation == '-':
    result = num1 - num2
    print("Result:", result)
elif operation == '*':
    result = num1 * num2
    print("Result:", result)
elif operation == '/':
    if num2 != 0:
        result = num1 / num2
        print("Result:", result)
    else:
        print("Error: Division by zero is not allowed!")
else:
    print("Invalid operation! Please enter +, -, *, or /.")
```

Output:

```
Enter first number: 12
Enter second number: 13
Select operation: + for addition, - for subtraction, * for multiplication, / for division
Enter operation: +
Result: 25.0
```

Q20. Write a python program to calculate the volume (V) of a cylinder given its radius(r) and height(h).

Formula: $V=\pi * r * r * h$

Solution:

```
# Import math module to use pi
import math

# Input radius and height from user
r = float(input("Enter the radius of the cylinder: "))
h = float(input("Enter the height of the cylinder: "))

# Calculate the volume
V = math.pi * r * r * h

# Print the volume
print("Volume of the cylinder is:", V)
```

Output:

```
Enter the radius of the cylinder: 4
Enter the height of the cylinder: 8
Volume of the cylinder is: 402.1238596594935
```

Q21. Write a python program to access the current year,month and date attributes.

Solution:

```
# Import datetime module
from datetime import datetime

# Get current date and time
current_date = datetime.now()

# Access year, month, and day
current_year = current_date.year
current_month = current_date.month
current_day = current_date.day

# Print the results
print("Current Year:", current_year)
print("Current Month:", current_month)
print("Current Day:", current_day)
```

Output:

```
Current Year: 2026
Current Month: 1
Current Day: 20
```

Q22. Implement python script to generate first N natural numbers.
Solution:

```
# Input N from the user
N = int(input("Enter the value of N: "))

# Generate first N natural numbers using a for loop
print("First", N, "natural numbers are:")
for i in range(1, N + 1):
    print(i, end=" ")  # Print numbers on the same line separated by space
```

Output:

```
Enter the value of N: 6
First 6 natural numbers are:
1 2 3 4 5 6
```

Q23. Implement python script to check given number is palindrome or not.

Solution:

```
# Input number from user
num = input("Enter a number: ")

# Check if the number is palindrome
if num == num[::-1]:
    print(num, "is a palindrome")
else:
    print(num, "is not a palindrome")
```

Output:

```
Enter a number: 67
67 is not a palindrome

== RESTART: C:/Users/zed ]
Enter a number: 66
66 is a palindrome
```

Q24. Write a python program that reads a positive integer from the user and computes the sum of its digits.

If input is :999 then output should be :27

If input is:640 then output should be :10

Solution:

```
# Input a positive integer
num = int(input("Enter a positive integer: "))

# Initialize sum
sum_of_digits = 0

# Compute sum of digits
while num > 0:
    digit = num % 10          # Get the last digit
    sum_of_digits += digit    # Add digit to sum
    num = num // 10           # Remove last digit

# Print the result
print("Sum of digits is:", sum_of_digits)
```

Output:

```
Enter a positive integer: 999
Sum of digits is: 27
```

Join our Telegram Channel

For better experience
DOWNLOAD
OUR APP

Success is the sum of small efforts... repeated day in and day out...