

Assignment No:08

Q.1.#WAP to calculate area of rectangle

```
def area(length,breadth):
    return length*breadth

x=int(input("Enter length:"))
y=int(input("Enter breadth:"))
output=area(x,y)
print("Area Of Rectangle=",output)
```

Q.2.#WAP to calculate area of circle

```
def cal_area(radius):
    area=3.14*radius**2
    return area

r=float(input("Enter radius of circle:"))
area=cal_area(r)
print("Area of circle is:",area)
```

Q.3.#write a program to find sum of following series using functions:

```
#a.1+2+3+4+.....+n
def sum_of_series(n):
    total=0
    for i in range(1,n+1):
        total+=i
    return total
n=int(input("Enter value of n:"))
output=sum_of_series(n)
print(f'The sum of the series 1+2+3+4+....+{n} is: {output}')
```

Q.3#write a program to find sum of following series using functions:

```
#b.1!+2!+3!+.....+n!
def factorial(num):
    fact=1
    for i in range(1,num+1):
        fact*=i
    return fact

def sum_of_factorial_series(n):
    total=0
    for i in range(1,n+1):
        total+=factorial(i)
    return total

n=int(input("Enter value of n:"))
output=sum_of_factorial_series(n)
print(f'The sum of the series 1!+2!+3!+.....+{n}!: {output}')
```

Q.3.#write a program to find sum of following series using functions:

```
#c.1^1+2^2+3^3+.....n^n
def sum_of_power_series(n):
    total=0
    for i in range(1,n+1):
        total+=i**i
```

```

    return total
n=int(input("Enter value of n:"))
output=sum_of_power_series(n)
print(f'The sum of series 1^1+2^2+3^3+.....{n}^{n}:{output} ')

```

Q.4.#WAP to print sum of all odd numbers between 1 to n

```

def sum_of_odd(n):
    total=0
    for i in range(1,n+1,2):
        total=total+i
    return total
n=int(input("Enter a number:"))
print("Sum of all odd numbers is",sum_of_odd(n))

```

Q.5.#WAP for sum of all prime numbers between 1 to n

```

def sumOfprime(n):
    sum=0
    for num in range(2,n+1):
        for i in range(2,(num//2)+1):
            if(num%i==0):
                break
        else:
            sum+=num
    return sum

```

```

print(sumOfprime(10))

```

Q.6.#WAP to find print the following fibonnacci series using function: 1 1 2 3 5 8 n terms

```

def fibonacci_series(n):
    a,b=1,1
    count=0
    while(count<n):
        print(a,end=' ')
        a,b=b,a+b
        count=count+1
n_term=int(input("Enter number of terms:"))
if(n_term<=0):
    print("Enter positive integer.")
else:
    print("Fibonacci series:")
    fibonacci_series(n_term)

```

Q.7.#WAP to find sum of digits of a number

```

def sum_of_digit(num):
    total=0
    while(num>0):
        d=num%10
        total=total+d
        num=num//10
    return total
x=int(input("Enter a number:"))
output=sum_of_digit(x)
print("sum of digit:",output)

```

Q.8.#WAP to find reverse of a number

```

def rev_digit(num):
    rev=0

```

```

while(num>0):
    d=num%10
    num=num//10
    rev=rev*10+d
return rev
number=int(input("Enter a number:"))
output=rev_digit(number)
print("Reverse number is:",output)

```

Q.9#wap to check if entered number is a palindrome or not

```

def is_palindrome(num):
    original_num=num
    reverse_num=0
    while(num>0):
        d=num%10
        reverse_num=(reverse_num*10) + d
        num//=10

    return original_num==reverse_num
number=int(input("Enter number:"))
if is_palindrome(number):
    print(f'{number} is a palindrome.')
else:
    print(f'{number} is not palindrome.')

```

Q.10.#wap to check if entered year is a leap year or not

```

def is_leap_year(year):
    if(year%4==0 and year%100!=0) or (year%400==0):
        return True
    else:
        return False
year_num=int(input("Enter year:"))
if is_leap_year(year_num):
    print(f'{year_num} is a leap year.')
else:
    print(f'{year_num} is not a leap year')

```

Q.11.#wap to check if a given number is Armstrong number or not. For each task create separate functions

```

def count_digits(num):
    count=0
    while(num>0):
        count+=1
        num=num//10
    return count

def armstrong_sum(num):
    power=count_digits(num)
    total=0
    for digit in str(num):
        total+=int(digit)**power
    return total

def is_armstrong(num):
    return num==armstrong_sum(num)

```

```
number=int(input("Enter number:"))

if is_armstrong(number):
    print(f'{number} is an Armstrong number.')
else:
    print(f'{number} is not an Armstrong number.')
```