

Functions

01) WAP to count simple interest using function.

```
def Interest(p,r,n):  
    ans=((p*r*n)/100)  
    print(ans)  
p=int(input('Enter p : '))  
r=int(input('Enter r : '))  
n=int(input('Enter n : '))  
Interest(p,r,n)
```

```
Enter p : 2  
Enter r : 3  
Enter n : 4  
0.24
```

02) WAP that defines a function to add first n numbers.

```
def add(n):  
    ans=(n*(n+1))/2  
    print(ans)  
n=int(input("Enter n : "))  
add(n)
```

```
Enter n : 5  
15.0
```

03) WAP to find maximum number from given two numbers using function.

```
def max(a,b):  
    if a>b:  
        print("a is a maximum number ")  
    else:  
        print("b is a maximum number ")  
a=int(input("Enter a : "))  
b=int(input("Enter b : "))  
max(a,b)
```

Enter a : 20
Enter b : 12
a is a maximum number

04) WAP that defines a function which returns 1 if the number is prime otherwise return 0.

```
def prime(n):  
    for i in range(2,n):  
        if(n%i==0):  
            print("This is a Not Prime")  
            return 0  
    else:  
        print("This is a Prime")  
        return 1  
n=int(input("Enter n : "))  
prime(n)
```

Enter n : 9
This is a Not Prime
0

05) Write a function called primes that takes an integer value as an argument and returns a list of all prime numbers up to that number.

```
def isprime(n):  
    for i in range(2,n):  
        if(n%i==0):  
            return 0  
    else:  
        return 1  
l1=[i for i in range(1,31)]  
ans=list(filter(isprime,l1))  
ans
```

[1, 2, 3, 5, 7, 11, 13, 17, 19, 23, 29]

06) WAP to generate Fibonacci series of N given number using function name fibbo. (e.g. 0 1 1 2 3 5 8...)

```
def Fibonacci(n):  
    if n<= 0:  
        print("Incorrect input")  
  
    elif n == 1:  
        return 0  
  
    elif n == 2:
```

```

        return 1
    else:
        return Fibonacci(n-1)+Fibonacci(n-2)
n=int(input("Enter n : "))
Fibonacci(n)

```

Enter n : 10

34

07) WAP to find the factorial of a given number using recursion.

```

def fact(n):
    f=1
    for i in range(1,n+1):
        f=f*i
    return f
l1=[i for i in range(1,n)]
ans=list(map(fact,l1))
ans
n=int(input("Enter n : "))
fact(n)

```

Enter n : 5

120

08) WAP to implement simple calculator using lamda function.

```

n1 = int(input("Enter Number : "))
n2 = int(input("Enter Number : "))
n3 = input("Enter Choice : ")
ans = lambda n1,n2,n3 : n1+n2 if n3=="+" else n1-n2 if n3=="-" else
n1*n2 if n3=="*" else n1//n2 if n3=="/" else "Wrong Operation"
print(ans(n1,n2,n3))

```

Enter Number : 10

Enter Number : 2

Enter Choice : /

5

09) Write a Python program that accepts a hyphen-separated sequence of words as input and prints the words in a hyphen-separated sequence after sorting them alphabetically

Sample Items : green-red-yellow-black-white Expected Result : black-green-red-white-yellow

```
str1 = input("Enter Hyphen Separated String : ")
list1 = str1.split("-")
list1.sort()
print("-".join(list1))
```

Enter Hyphen Separated String : green-red-yellow-black-white
black-green-red-white-yellow

10) Write a python program to implement all function arguments type

Positional arguments Default argument Keyword arguments (named arguments)
Arbitrary arguments (variable-length arguments args and kwargs)

```
a = int(input("Enter Number : "))
b = int(input("Enter Number : "))
positionalArguments = lambda a,b : a+b
print("Positional arguments = ",positionalArguments(a,b))
```

```
a = int(input("Enter Number : "))
defaultArgument = lambda a,b=10 : a+b
print("Default argument = ",defaultArgument(a))
```

Enter Number : 2
Enter Number : 4
Positional arguments = 6
Enter Number : 5
Default argument = 15

```
a = int(input("Enter Number : "))
b = int(input("Enter Number : "))
def keywordArguments(a,b):
    return a+b
print("Keyword arguments (named arguments) = ",keywordArguments(b=a,a=b))
```

```
a = int(input("Enter Number : "))
def arbitraryArguments(a,*b):
    sums = a
    for i in b:
        sums+=i
    return sums
print("Keyword arguments (named arguments) = ",arbitraryArguments(a,5,10))
```

```
Enter Number : 5
Enter Number : 10
Keyword arguments (named arguments) = 15
Enter Number : 4
Keyword arguments (named arguments) = 19
```

01) WAP to calculate power of a number using recursion.

```
def powers(base,power):
    if power==1:
        return base
    elif power==0:
        return 1
    else:
        return base*powers(base,power-1)
base = int(input("Enter base : "))
power = int(input("Enter power : "))
powers(base,power)
```

```
Enter base : 3
Enter power : 3
```

27

02) WAP to count digits of a number using recursion.

```
def countDigits(n):
    if n<10:
        return 1
    else:
        return 1+countDigits(n//10)
```

```
n = int(input("Enter Number : "))
ans = countDigits(n)
print(ans)
```

```
Enter Number : 23457645
8
```

03) WAP to reverse an integer number using recursion.

```
reverse_number = 0
def reverseNumber(n):
    global reverse_number
    if(n>0):
        reminder = n%10
        reverse_number = reverse_number*10 + reminder
        reverseNumber(n//10)
    return reverse_number
```

```
n = int(input("Enter Number : "))
```

```
reverse_number = reverseNumber(n)
print("Reverse : ",reverse_number)
```

```
Enter Number : 12345
Reverse : 54321
```

04) WAP to convert decimal number into binary using recursion.

```
def decimalToBinary(n):
    if n==0:
        return 0
    else:
        return n%2+10*(decimalToBinary(n//2))
```

```
n = int(input("Enter Number : "))
ans = decimalToBinary(n)
print("Binary : ",ans)
```

```
Enter Number : 16
Binary : 10000
```