Darshan UNIVERSITY

योग: कर्मसु कौशलम्

Python Programming - 2101CS405

Lab - 7

Functions

In [ ]:
```
Name :- Vora Yagnik
Enrollment No :- 23010101661
```

01) WAP to count simple interest using function.

SI =(principle(amount) * rate of interest * time)/ 100

In [3]:
```
def interest(p,r,n):
    return (p*r*n)/100

interest(100000,2,2)
```

Out[3]: 4000.0

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

In [7]:
```
def sumN(n):
    sum = 0
    for i in range(1,n+1):
        sum += i
    return sum

n = int(input("Enter N : "))
print("Sum of first ",n," = ",sumN(n))
```

```
Sum of first  10  =  55
```

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

In [12]:
```
def maxOfTwo(a,b):
    return a if a>b else b

a = int(input("Enter first number : "))
b = int(input("Enter second number : "))

print("Max from",a,"and",b,"is",maxOfTwo(a,b))
```

```
Max from 10 and 20 is 20
```

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

In [43]:
```
def isPrime(n):
    c = 0
    for i in range(2,int(n**0.5)):
        c += 1
        if n%i == 0:
            return 0,c
    return 1,c
n = int(input("Enter number : "))
prime,iteration = isPrime(n)
print("No. of iterations : ",iteration,"\n",prime)
```

```
No. of iterations :  4
 0
```

In [1]:
```
# Example on how iterations are calculated
# Only iteration in loops are counted
iteration_count= 0
n = 5
for i in range(n):
    iteration_count += 1

    for j in range(n-i-1):
        iteration_count += 1
        print(" ", end = " ")
    for j in range(i+1):
        iteration_count += 1
        print("*", end = " ")
    print()
print("Total Iterations : ", iteration_count)
```

```
      *
     * *
    * * *
   * * * *
* * * * *
Total Iterations :  30
```

In [20]:
```

```

```
Enter Number :  9973
No. of iteration :  16
1
```

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.

In [83]:
```python
def primes(n):
    p = []
    t,c = 0,0
    for i in range(2,n+1):
        t += c
        c = 0
        f = True
        for j in range (2,i):
            c += 1
            if i%j == 0:
                f = False
        if f==True:
            p.append(i)
    return p,t
i,c = primes(1000)
print(c)
```

```
497503
```

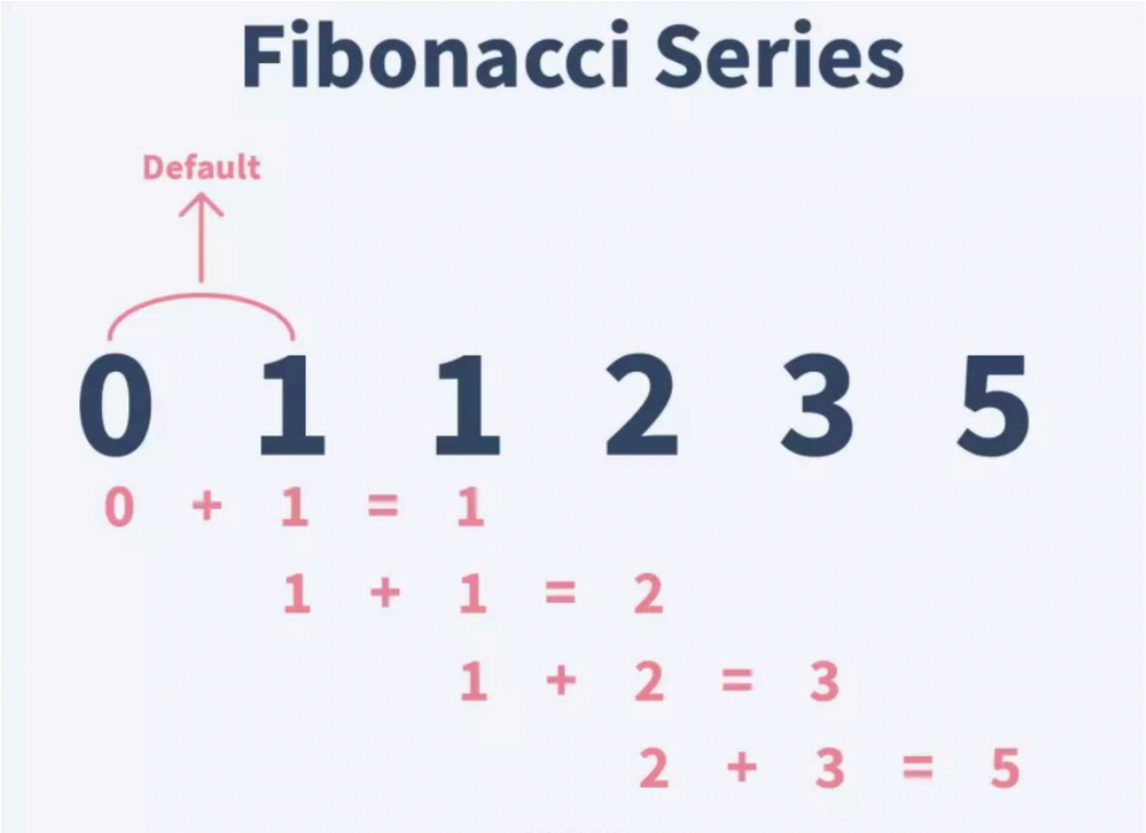
In [93]:
```python
c = 0
def primes(n):
    global c
    for i in range(2,n):
        for j in range (2,int(i**0.5)):
            c += 1
            if i%j == 0:
                break
        else:
            print(i,end=" ")
print(primes(1000))
print(c)
```

```
2 3 4 5 6 7 8 9 11 13 15 17 19 23 25 29 31 35 37 41 43 47 49 53 59 61 67 71 73 79 83
89 97 101 103 107 109 113 121 127 131 137 139 143 149 151 157 163 167 169 173 179 18
1 191 193 197 199 211 223 227 229 233 239 241 251 257 263 269 271 277 281 283 289 29
3 307 311 313 317 323 331 337 347 349 353 359 361 367 373 379 383 389 397 401 409 41
9 421 431 433 439 443 449 457 461 463 467 479 487 491 499 503 509 521 523 529 541 54
7 557 563 569 571 577 587 593 599 601 607 613 617 619 631 641 643 647 653 659 661 67
3 677 683 691 701 709 719 727 733 739 743 751 757 761 769 773 787 797 809 811 821 82
3 827 829 839 841 853 857 859 863 877 881 883 887 899 907 911 919 929 937 941 947 95
3 961 967 971 977 983 991 997 None
5103
```

optional task : optimization

if basic task completed then try to optimize

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



In [10]:
```
def fibo(n):
    a = 0
    b = 1
    print(a,b,end=",",sep=",")
    for i in range(2,n):
        c = a+b
        print(c,end=",")
        a = b
        b = c

n = int(input("Enter no of Steps : "))
fibo(n)
```

```
0,1,1,2,3,5,8,13,21,34,
```

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

In [19]:
```
def fact(n):
    if n == 0:
        return 1
    else:
        return n*fact(n-1)
```

```
n = int(input("Enter number to find factorial : "))
fact(n)
```

Out[19]: 5040

08) WAP to implement simple calculator using lamda function.

In [25]:
```
a = int(input("Enter first no :- "))
b = int(input("Enter second no :- "))
c = input("Enter Operation [+,-,*,/] :- ")

if c == '+':
    print("Ans :- ",(lambda a,b:a+b)(a,b))
elif c == '-':
    print("Ans :- ",(lambda a,b:a-b)(a,b))
elif c == '*':
    print("Ans :- ",(lambda a,b:a*b)(a,b))
elif c == '/':
    print("Ans :- ",(lambda a,b:a/b)(a,b))
```

```
Ans :-  10
```

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

In [31]:
```
s = "green-red-yellow-black-white"
l = s.split("-")
l = sorted(l)
print("-".join(l))
```

```
black-green-red-white-yellow
```

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

In [34]:
```
def powerD(base,power):
    if power == 0:
        return 1
    else:
        return base*powerD(base,power-1)

powerD(3,3)
```

Out[34]: 27

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

In [42]:
```
def countD(n):
    if n//10 == 0:
        return 1
    else:
        return 1+countD(n//10)
countD(1418150)
```

Out[42]: 7

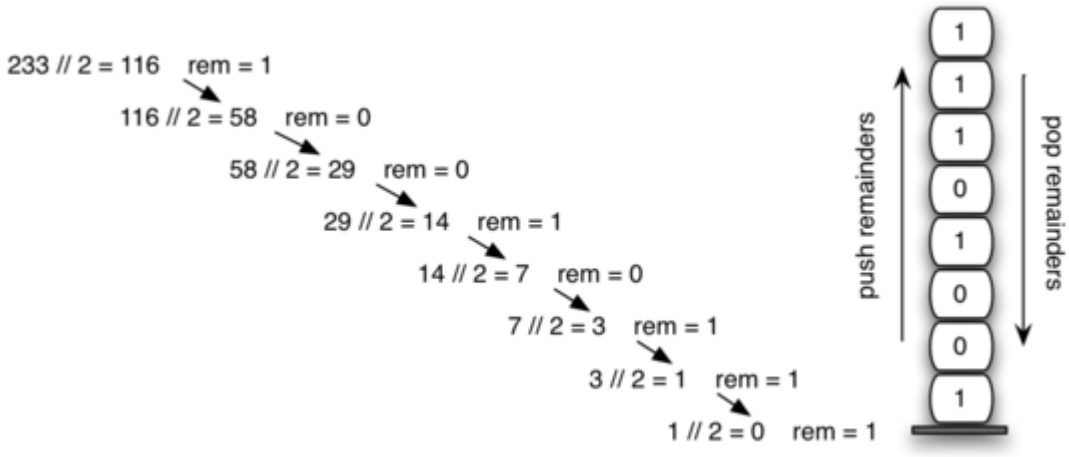
03) WAP to reverse an integer number using recursion.

In [44]:
```
def reverseN(n,rev_n=0):
    if n ==0:
        return rev_n
    else:
        lDigit = n%10
        rev_n = rev_n*10 + lDigit
        return reverseN(n//10,rev_n)

reverseN(2151515)
```

Out[44]: 5151512

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



Decimal To Binary Converstion:

Let the decimal number be : 14

2	14	0
2	7	1
2	3	1
2	1	1
	0	

$(14)_{10} = (1110)_2$

Let the decimal number be : 22

2	22	0
2	11	1
2	5	1
2	2	0
2	1	1
	0	

$(21)_{10} = (10110)_2$

In [55]:
```
def dtob(n):
    str = ""
    if n == 0:
        return 0
    else:
        return (n%2 + 10*dtob(int(n//2)))
dtob(10)
```

Out[55]: 1010

Map , Filter , Reduce

map() function returns a map object(which is an iterator) of the results after applying the given function to each item of a given iterable (list, tuple etc.)

In [3]:
```
numbers = [1, 2, 3, 4, 5]

squared_numbers = list(map(lambda x: x * x, numbers))

print(squared_numbers)
```

```
[1, 4, 9, 16, 25]
```

The filter() method filters the given sequence with the help of a function that tests each element in the sequence to be true or not.

In [1]:
```
seq = [0, 1, 2, 3, 5, 8, 13]
```

```
result = filter(lambda x: x % 2 != 0, seq)
print(list(result))

result = filter(lambda x: x % 2 == 0, seq)
print(list(result))
```

```
[1, 3, 5, 13]
[0, 2, 8]
```

The reduce(fun,seq) function is used to apply a particular function passed in its argument to all of the list elements mentioned in the sequence passed along.This function is defined in “functools” module.

In [2]:
```
import functools

lis = [1, 3, 5, 6, 2]

print("The sum of the list elements is : ", end="")
print(functools.reduce(lambda a, b: a+b, lis))

print("The maximum element of the list is : ", end="")
print(functools.reduce(lambda a, b: a if a > b else b, lis))
```

```
The sum of the list elements is : 17
The maximum element of the list is : 6
```