

Python Programs for Beginners :

1. Using for loop, print a table of Celsius/Fahrenheit equivalences. Let c be the Celsius temperatures ranging from 0 to 100, for each value of c, print the corresponding Fahrenheit temperature.
2. Using while loop, produce a table of sines, cosines and tangents. Make a variable x in range from 0 to 10 in steps of 0.2. For each value of x, print the value of $\sin(x)$, $\cos(x)$ and $\tan(x)$.
3. Write a program that reads an integer value and prints "leap year" or "not a leap year".
4. Write a program that takes a positive integer n and then produces n lines of output shown as follows. For example, enter a size: 4 * ** *** ****
5. Write a function that takes an integer 'n' as input and calculates the value of $1 + 1/1! + 1/2! + 1/3! + \dots + 1/n$
6. Write a function that takes an integer input and calculates the factorial of that number.
7. Write a function that takes a string input and checks if it's a palindrome or not.
8. Write a list function to convert a string into a list, as in list ('abc') gives [a, b, c].
9. Write a program to generate Fibonacci series.
10. Write a program to check whether the input number is even or odd.
11. Write a program to compare three numbers and print the largest one.
12. Write a program to print factors of a given number.
13. Write a method to calculate GCD of two numbers.
14. Write a program to create Stack Class and implement all its methods. (Use Lists)
15. Write a program to create Queue Class and implement all its methods. (Use Lists)
16. Write a program to implement linear and binary search on lists.
17. Write a program to sort a list using insertion sort and bubble sort and selection

1. Using for loop, print a table of Celsius/Fahrenheit equivalences.

For example: Let c be the Celsius temperatures ranging from 0 to 100, for each value of c, print the corresponding Fahrenheit temperature.

```
# creating a function that converts the given celsius degree temperature to Fahrenheit temperature
def convertCelsiustoFahrenheit(c):
    # converting celsius degree temperature to Fahrenheit temperature
    f = (9/5)*c + 32
    # returning Fahrenheit temperature of given celsius temperature
    return (f)

for ctemp in range(1, 101):
    ftemp = convertCelsiustoFahrenheit(ctemp)
    #print("The Fahrenheit equivalent of ", ctemp , "is", ftemp)
    print("The Fahrenheit equivalent of ", ctemp , "celsius is", "%.2f" % ftemp)
```

```
The Fahrenheit equivalent of 1 celsius is 33.80
The Fahrenheit equivalent of 2 celsius is 35.60
The Fahrenheit equivalent of 3 celsius is 37.40
The Fahrenheit equivalent of 4 celsius is 39.20
```

And so on ----- upto

```
The Fahrenheit equivalent of 99 celsius is 210.20
The Fahrenheit equivalent of 100 celsius is 212.00
```

2. Using while loop, produce a table of sins, cosines and tangents. Make a variable x in range from 0 to 10 in steps of 0.2. For each value of x, print the value of sin(x), cos(x) and tan(x).

```
# Import math Library
import math

x = 0
while x <= 10:
    print("x is equal to ", round(x,1))
    print("sin x is ", round(math.sin(x),4))
    print("cos x is ",round(math.cos(x),4))
    print("tan x is ",round(math.tan(x),4))
    print()
    x = x + 0.2
```

```
x is equal to 0
sin x is 0.0
cos x is 1.0
tan x is 0.0

x is equal to 0.2
sin x is 0.1987
cos x is 0.9801
tan x is 0.2027
```

And so on ----- upto

```
x is equal to 10.0
sin x is -0.544
cos x is -0.8391
tan x is 0.6484
```

3. Write a program that reads an integer value and prints "leap year" or "not a leap year".

```
# To get year (integer input) from the user
year = int(input("Enter a year: "))

# divided by 100 means century year (ending with 00)
# century year divided by 400 is leap year
if (year % 400 == 0) and (year % 100 == 0):
    print("{0} is a leap year".format(year))

# not divided by 100 means not a century year
# year divided by 4 is a leap year
elif (year % 4 == 0) and (year % 100 != 0):
    print("{0} is a leap year".format(year))

# if not divided by both 400 (century year) and 4 (not century year)
# year is not leap year
else:
    print("{0} is not a leap year".format(year))
```

```
Enter a year: 2000
2000 is a leap year
```

4. Write a program that takes a positive integer n and then produces n lines of output shown as follows.

```

n = int(input("Enter the number of lines "))
# outer loop to handle number of lines
for i in range(0, n):
    # inner loop to handle number of columns
    # values is changing according to outer loop
    for j in range(0, i + 1):
        # printing @
        print("* ", end="")

    # ending line after each row
    print()

```

```

Enter the number of lines 6
*
* *
* * *
* * * *
* * * * *
* * * * *

```

5. Write a function that takes an integer 'n' as input and calculates the value of $1 + 1/1! + 1/2! + 1/3! + \dots + 1/n$

```

def factorial(n):

    res = 1
    for i in range(2,n+1):
        res =res*i
    return res

# A Simple Function to return value of 1/1! + 1/2! + .. + 1/n!
def sum(n):

    sum = 0
    for i in range (1, n+1):
        sum =sum+ 1/factorial(i)
    return sum

#n = 5
#input the value of n
n = int(input("enter the term number"))

print("The sum of series 1/1! + 1/2! + 1/3! + 1/4! .... upto the term ", n)
print("factorial of the nth term is" , factorial(n))
#print("The sum of series 1/1! + 1/2! + 1/3! + 1/4! .... is ", sum(n))
print("The sum of series 1/1! + 1/2! + 1/3! + 1/4! .... is {:.2f}".format(sum(n)))

```

```
enter the term number 5
The sum of series 1/1! + 1/2! + 1/3! + 1/4! .... upto the term 5
factorial of the nth term is 120
The sum of series 1/1! + 1/2! + 1/3! + 1/4! .... is 1.72
```

6. Write a function that takes an integer input and calculates the factorial of that number.

```
# Program to print factorial of a number recursively.

# Recursive function
def recursive_factorial(n):
    if n == 1:
        return n
    else:
        return n * recursive_factorial(n-1)

#input the value of n
num = int(input("Enter value of n: "))

# check if the input is valid or not
if num < 0:
    print("Invalid input ! Please enter a positive number.")
elif num == 0:
    print("Factorial of number 0 is 1")
else:
    print("Factorial of number", num, "=", recursive_factorial(num))
```

```
Enter value of n: 4
Factorial of number 4 = 24
```

7. Write a function that takes a string input and checks if it's a palindrome or not.

```
# function which return reverse of a string

def isPalindrome(s):
    print(s)
    print(s[::-1])
    return s == s[::-1]

# Driver code
#s = "malayalam"
s = input("Enter a string to check if it is a palindrome: ")
ans = isPalindrome(s)

if ans:
    print("Yes, it is a palindrome")
else:
    print("No, it is not a palindrome")
```

```
Enter a string to check if it is a palindrome: malayalam
malayalam
malayalam
Yes, it is a palindrome
```

8. Write a list function to convert a string into a list, as in list ('abc') gives [a, b, c].

```
s=input("Enter a string here: ")
x=[i for i in s]
print(x)
```

```
Enter a string here: Hello World
['H', 'e', 'l', 'l', 'o', ' ', 'W', 'o', 'r', 'l', 'd']
```

9. Write a program to generate Fibonacci series.

```

# Program to display the Fibonacci sequence up to n-th term

nterms = int(input("How many terms? "))

# first two terms
n1, n2 = 0, 1
count = 0

# check if the number of terms is valid
if nterms <= 0:
    print("Please enter a positive integer")
# if there is only one term, return n1
elif nterms == 1:
    print("Fibonacci sequence upto",nterms,":")
    print(n1)
# generate fibonacci sequence
else:
    print("Fibonacci sequence:")
    while count < nterms:
        print(n1)
        nth = n1 + n2
        # update values
        n1 = n2
        n2 = nth
        count += 1

```

```

How many terms? 7
Fibonacci sequence:
0
1
1
2
3
5
8

```

10. Write a program to check whether the input number is even or odd.

```

num = int (input ("Enter any number to test whether it is odd or even: "))

if (num % 2) == 0:
    print ("The number is even")
else:
    print ("The provided number is odd")

```

```
Enter any number to test whether it is odd or even: 3
The provided number is odd
```

11. Write a program to compare three numbers and print the largest one.

```
num1=input("Enter a number num1 : ")
num2=input("Enter a number num2 : ")
num3=input("Enter a number num3 : ")

max = 0
if num1 >= num2 and num1 >= num3:
    print(num1)
elif num2 >= num1 and num2 >= num3:
    print(num2)
else:
    print(num3)
```

```
Enter a number num1 : 7
Enter a number num2 : 5
Enter a number num3 : 6
7
```

12. Write a program to print factors of a given number.

```
# Python Program to find the factors of a number

# This function computes the factor of the argument passed
def print_factors(x):
    print("The factors of",x,"are:")
    for i in range(1, x + 1):
        if x % i == 0:
            print(i)

#num = 320
num=int(input("Enter a number : "))
print_factors(num)
```



```
Enter a number : 100
The factors of 100 are:
1
2
4
5
10
20
25
50
100
```

13. Write a method to calculate GCD of two numbers.

```
def gcd(x,y):
    if x == y or y == 0:
        return x
    if x == 0:
        return y

    else:
        if x>y:
            return gcd(x-y,y)
        else:
            return gcd(x,y-x)

print(" The gcd of the given numbers is ", gcd(27,90))
```

```
The gcd of the given numbers is 9
```

14. Write a program to create Stack Class and implement all its methods.
(Use Lists)

```

class stack:
    def __init__(self):
        self.__index = []

    def __len__(self):
        return len(self.__index)

    def push(self,item):
        self.__index.insert(0,item)

    def peek(self):
        if len(self) == 0:
            raise Exception("peek() called on empty stack.")
        return self.__index[0]

    def pop(self):
        if len(self) == 0:
            raise Exception("pop() called on empty stack.")
        return self.__index.pop(0)

    def __str__(self):
        return str(self.__index)

s=stack()
print(s)
s.push(5)
s.push('apple')
s.push({'keyA': 'valueb'})
print(s)
s.peek()
print(len(s))
s.pop()
print(s)

```

```

[]
[{'keyA': 'valueb'}, 'apple', 5]
3
['apple', 5]

```

15. Write a program to create Queue Class and implement all its methods. (Use Lists)

```

# Implement Queue using List(Class Concept)
class Queue:
    def __init__(self):
        self.__index = []

    def lengthofQ(self):
        len(self.__index)
        print(len(self.__index) ,"is the length of the Queue!")

    def Enqueue(self,item):
        self.__index.append(item)
        print(item,"is added to the Queue!")

    def dequeue(self):
        if not self.__index:
            print("Queue is Empty!!!")
        else:
            e=self.__index.pop(0)
            print("element removed!!:",e)

    def __str__(self):
        return str(self.__index)

q=Queue()
q.Enqueue(2)
q.Enqueue(3)
q.Enqueue(4)
q.Enqueue(10)
q.Enqueue(11)
q.Enqueue(12)
q.lengthofQ()
print(q)
q.dequeue()
print(q)

```

```

2 is added to the Queue!
3 is added to the Queue!
4 is added to the Queue!
10 is added to the Queue!
11 is added to the Queue!
12 is added to the Queue!
6 is the length of the Queue!
[2, 3, 4, 10, 11, 12]
element removed!!: 2
[3, 4, 10, 11, 12]

```

16. Write a program to implement linear and binary search on lists.

Linear Search:

```
def search(List, n):  
  
    for i in range(len(List)):  
        if List[i] == n:  
            return True  
    return False  
  
# this list contains both string and numbers.  
List = [1, 2, 'name', 4, 'This', 6]  
  
#try these to check one by one  
#n = 4  
#n="This"  
n="this"
```

Not Found

Binary Search:

```

def binary_search(arr, x):
    low = 0
    high = len(arr) - 1
    mid = 0

    while low <= high:

        mid = (high + low) // 2

        # If x is greater, ignore left half
        if arr[mid] < x:
            low = mid + 1

        # If x is smaller, ignore right half
        elif arr[mid] > x:
            high = mid - 1

        # means x is present at mid
        else:
            return mid

    # If we reach here, then the element was not present
    return -1

# Test array
arr = [ 2, 3, 4, 10, 40 ]
x = 3

# Function call
result = binary_search(arr, x)

if result != -1:
    print("Element is present at index", str(result))
else:
    print("Element is not present in array")

```

Element is present at index 1

17. Write a program to sort a list using insertion sort and bubble sort and selection

Insertion sort :

```
def insertionSort(arr):
    n = len(arr) # Get the length of the array

    if n <= 1:
        return # If the array has 0 or 1 element, it is already sorted, so return

    for i in range(1, n): # Iterate over the array starting from the second element
        key = arr[i] # Store the current element as the key to be inserted in the right position
        j = i-1
        while j >= 0 and key < arr[j]: # Move elements greater than key one position ahead
            arr[j+1] = arr[j] # Shift elements to the right
            j -= 1
        arr[j+1] = key # Insert the key in the correct position

# Sorting the array [12, 11, 13, 5, 6] using insertionSort
arr = [12, 71, 13, 5, 6]
insertionSort(arr)
print(arr)
```

[5, 6, 12, 13, 71]

Bubble sort :

```
# Python program for implementation of Bubble Sort

def bubbleSort(arr):
    n = len(arr)
    # optimize code, so if the array is already sorted, it doesn't need
    # to go through the entire process
    swapped = False
    # Traverse through all array elements
    for i in range(n-1):
        # range(n) also work but outer loop will
        # repeat one time more than needed.
        # Last i elements are already in place
        for j in range(0, n-i-1):

            # traverse the array from 0 to n-i-1
            # Swap if the element found is greater
            # than the next element
            if arr[j] > arr[j + 1]:
                swapped = True
                arr[j], arr[j + 1] = arr[j + 1], arr[j]

        if not swapped:
            # if we haven't needed to make a single swap, we
            # can just exit the main loop.
            return

# Driver code to test above
arr = [64, 34, 25, 12, 22, 11, 90]

bubbleSort(arr)

print("Sorted array is:")
for i in range(len(arr)):
    print("% d" % arr[i], end=" ")
```

```
Sorted array is:
11 12 22 25 34 64 90
```