

PYTHON ASSIGNMENT FUNCTION

1. Write a Python function to find the maximum of three numbers.

```
def max_of_two( x, y ):
    if x > y:
        return x
    return y
def max_of_three( x, y, z ):
    return max_of_two( x, max_of_two( y, z ) )
print(max_of_three(3, 6, -5))
```

2. Write a Python function to calculate the factorial of a number (a non-negative integer). The function accepts the number as an argument.

```
def factorial(n):
    if n == 0:
        return 1
    else:
        return n * factorial(n-1)
n=int(input("Input a number to compute the factiorial : "))
print(factorial(n))
```

3. Write a Python function to check whether a number falls within a given range.

```
def test_range(n):
    if n in range(3,9):
        print( " %s is in the range"%str(n))
    else :
        print("The number is outside the given range.")
test_range(5)
```

4. Write a Python function that accepts a string and counts the number of upper and lower case letters.

```
def string_test(s):  
    d={"UPPER_CASE":0, "LOWER_CASE":0}  
    for c in s:  
        if c.isupper():  
            d["UPPER_CASE"]+=1  
        elif c.islower():  
            d["LOWER_CASE"]+=1  
        else:  
            pass  
    print ("Original String : ", s)  
    print ("No. of Upper case characters : ", d["UPPER_CASE"])  
    print ("No. of Lower case Characters : ", d["LOWER_CASE"])  
  
string_test('DiSHA Computer Institute Dange Chowk')
```

5. Write a Python function that takes a number as a parameter and checks whether the number is prime or not.

Note : A prime number (or a prime) is a natural number greater than 1 and that has no positive divisors other than 1 and itself.

Solution 1:

```
def test_prime(n):  
    if (n==1):  
        return False  
    elif (n==2):  
        return True;  
    else:  
        for x in range(2,n):  
            if(n % x==0):
```

```
        return False
    return True
print(test_prime(9))
```

Solution 2:

```
from math import sqrt
def is_prime(num):
    if num <= 1 or (num % 2 == 0 and num > 2):
        return False
    return all(num % i for i in range(3, int(sqrt(num)) + 1, 2))
print(is_prime(11))
print(is_prime(13))
print(is_prime(16))
print(is_prime(17))
print(is_prime(97))
```

Lambda Function

1. Write a Python program to create a function that takes one argument, and that argument will be multiplied with an unknown given number.

```
def func_compute(n):
    return lambda x : x * n
result = func_compute(2)
print("Double the number of 15 =", result(15))
result = func_compute(3)
print("Triple the number of 15 =", result(15))
result = func_compute(4)
print("Quadruple the number of 15 =", result(15))
```

```
result = func_compute(5)
print("Quintuple the number 15 =", result(15))
```

2. Write a Python program to sort a list of tuples using Lambda.

```
subject_marks = [('English', 88), ('Science', 90), ('Maths', 97), ('Social sciences', 82)]
print("Original list of tuples:")
print(subject_marks)
subject_marks.sort(key = lambda x: x[1])
print("\nSorting the List of Tuples:")
print(subject_marks)
```

3. Write a Python program to sort a list of dictionaries using Lambda.

```
models = [{'make':'Nokia', 'model':216, 'color':'Black'}, {'make':'Mi Max', 'model':2, 'color':'Gold'}, {'make':'Samsung', 'model': 7, 'color':'Blue'}]
print("Original list of dictionaries :")
print(models)
sorted_models = sorted(models, key = lambda x: x['color'])
print("\nSorting the List of dictionaries :")
print(sorted_models)
```

4. Write a Python program to filter a list of integers using Lambda.

```
nums = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
print("Original list of integers:")
print(nums)
print("\nEven numbers from the said list:")
```

```
even_nums = list(filter(lambda x: x%2 == 0, nums))
print(even_nums)
print("\nOdd numbers from the said list:")
odd_nums = list(filter(lambda x: x%2 != 0, nums))
print(odd_nums)
```

5. Write a Python program to find whether a given string starts with a given character using Lambda.

```
starts_with = lambda x: True if x.startswith('P') else False
print(starts_with('Python'))
starts_with = lambda x: True if x.startswith('P') else False
print(starts_with('Java'))
```

6. Write a Python program to extract year, month, date and time using Lambda.

```
import datetime
now = datetime.datetime.now()
print(now)
year = lambda x: x.year
month = lambda x: x.month
day = lambda x: x.day
t = lambda x: x.time()
print(year(now))
print(month(now))
print(day(now))
print(t(now))
```

7. Write a Python program to filter a given list to determine if the values in the list have a length of 6 using Lambda.

```
weekdays = ['Monday', 'Tuesday', 'Wednesday', 'Thursday', 'Friday', 'Saturday', 'Sunday']  
  
days = filter(lambda day: day if len(day)==6 else "", weekdays)  
  
for d in days:  
    print(d)
```

8. Write a Python program to add two given lists using map and lambda.

```
nums1 = [1, 2, 3]  
nums2 = [4, 5, 6]  
print("Original list:")  
print(nums1)  
print(nums2)  
result = map(lambda x, y: x + y, nums1, nums2)  
print("\nResult: after adding two list")  
print(list(result))
```

9. Write a Python program that multiplies each number in a list with a given number using lambda functions. Print the results.

```
nums = [2, 4, 6, 9, 11]  
n = 2  
print("Original list: ", nums)  
print("Given number: ", n)  
filtered_numbers=list(map(lambda number:number*n,nums))  
print("Result:")  
print(' '.join(map(str,filtered_numbers)))
```

10. Write a Python program to find the index position and value of the maximum and minimum values in a given list of numbers using lambda.

```
def position_max_min(nums):  
    max_result = max(enumerate(nums), key=(lambda x: x[1]))  
    min_result = min(enumerate(nums), key=(lambda x: x[1]))  
    return max_result, min_result  
  
nums = [12, 33, 23, 10.11, 67, 89, 45, 66.7, 23, 12, 11, 10.25, 54]  
print("Original list:")  
print(nums)  
result = position_max_min(nums)  
print("\nIndex position and value of the maximum value of the said list:")  
print(result[0])  
print("\nIndex position and value of the minimum value of the said list:")  
print(result[1])
```

11. Write a Python program to find the elements of a given list of strings that contain a specific substring using lambda.

```
def find_substring(str1, sub_str):  
    result = list(filter(lambda x: sub_str in x, str1))  
    return result  
  
colors = ["red", "black", "white", "green", "orange"]  
print("Original list:")  
print(colors)  
  
sub_str = "ack"
```

```
print("\nSubstring to search:")  
print(sub_str)  
print("Elements of the said list that contain specific substring:")  
print(find_substring(colors, sub_str))  
sub_str = "abc"  
print("\nSubstring to search:")  
print(sub_str)  
print("Elements of the said list that contain specific substring:")  
print(find_substring(colors, sub_str))
```