WEEK – 8

1. Write a python program to find factorial of a number using Recursion.

Source code:-

```
def recur_factorial(n):
   if n == 1:
       return n
   else:
       return n*recur_factorial(n-1)

num = 7

# check if the number is negative
if num < 0:
   print("Sorry, factorial does not exist for negative numbers")
elif num == 0:
   print("The factorial of 0 is 1")
else:
   print("The factorial of", num, "is", recur_factorial(num))
```

Output:-

```
The factorial of 7 is 5040
```

Description:-

Here, the number is stored in `num`. The number is passed to the `recur_factorial()` function to compute the factorial of the number.

2. Write a program that accepts the lengths of three sides of a triangle as inputs. The program output should indicate whether or not the triangle is a right angled triangle (Recall from the Pythagorean Theorem that in a right angled triangle, the square of one side equals the sum of the squares of the other two sides).

Source code:-

```
base=float(input("Enter length of Base : "))
perp=float(input("Enter length of Perpendicular : "))
hypo=float(input("Enter length of Hypotenuse : "))

if hypo**2==((base**2)+(perp**2)):
    print("It's a right triangle")
else:
    print("It's not a right triangle")
```

Output:-

```
E:\Python>python week13.py
Enter length of Base : 3
Enter length of Perpendicular : 4
Enter length of Hypotenuse : 5
It's a right triangle

E:\Python>python week13.py
Enter length of Base : 2
Enter length of Perpendicular : 3
Enter length of Hypotenuse : 4
It's not a right triangle
```

Description:-

Pythagoras theorem says: In a right-angled triangle, the square of the hypotenuse side is equal to the sum of squares of the other two sides.

3. Write a Python function to multiply all the numbers in a list

Source code:-

```
def mul_list(list) :

    # Multiply elements one by one
    product = 1
    for i in list:
         product = product * i
    return product


list1 = [11, 12, 4, 3]
print(list1)
print("product: ")
print(mul_list(list1))
```

Output:-

```
[11, 12, 4, 3]

product:

1584
```

Description:-

In this approach, we will traverse till the end of the list to find the product. Initialize a variable product to 1 and multiply each number in the list with a product to get the result. To access each number in the list we will use for loop in Python.

- Define a function to multiply numbers
- Declare a variable product and set it to 1
- Run a loop for all elements in the list
- Multiply all elements to the product
- Return product
- Declare a list
- Pass list in our function
- Print value returned by the function

4. Write a python program to define a module to find Fibonacci Numbers and import the module to another program.

Source code:-

fibonacci.py

```
# Fibonacci numbers module

def fib(n):    # write Fibonacci series up to n
    a, b = 0, 1
    while b < n:
        print(b, end =" ")
        a, b = b, a+b
```

Main.py

```
#import fibonacci module
import fibonacci
num=int(input("Enter any number to print Fibonacci series "))
fibonacci.fib(num)
```

Output:-

```
E:\Python>python week14.py
Enter any number to print Fibonacci series 50
1 1 2 3 5 8 13 21 34
E:\Python>python week14.py
Enter any number to print Fibonacci series 100
1 1 2 3 5 8 13 21 34 55 89
```

Description:-

To use the one module in another program, we can import it using the import statement. Here in this program that imports the Fibonacci module and uses it to print the Fibonacci numbers.