1. Factorial of a given number, example shown below

Graphical user interface, text, application

Description automatically generated

To find the factorial of a given number we need to start with 1 and end with the given number:

As of now let’s assign x = 5

X = 5

Result = Fact(5)

Print(result)

Now let’s define the function as shown below:

Def fact(n):

F =1 # we will start multiplying starting 1

For I in range(1,n+1): # here n + 1 given because if we give 5 it will stop at 4 so we need 5 as well, hence we used n+1

F = f \* i

Return f

🡪debug and see how it works, play with return.

def fact(n):

    f = 1

    for i in range(1,n+1):

        f = f \* i

    return f

x = 5

result = fact(5)

print(result)

Recursion – calling a function from itself

Function will call itself infinite times and it will give an error after printing some lines: bydefault the limit is 1000

Before recursion :

Timeline

Description automatically generated with low confidence

After recursion: what’s happening here is the function calling itself and python will call itself 1000 times by default which will save the resources

Graphical user interface, application

Description automatically generated with medium confidence

* To know how many times a recursion occurs we need to run something like this:

Diagram

Description automatically generated with medium confidence

* If we want to change the recursion limit, we can do it. Instead of 2000 we can set a different so that the recursion stops at 2000.
* Graphical user interface, text, application, email

  Description automatically generated

Now to check how many times we are printing hello we will use a counter variable I = 0. Inside the for loop we will make the inside variable a global and increment it.

Graphical user interface, application

Description automatically generated

import sys

sys.setrecursionlimit(2000)

print(sys.getrecursionlimit())

i = 0

def greet():

    global i

    i += 1

    print('hello', i)

    greet()

greet()

\*\* factorial using recursion

Result = Fact(5)

Print(result)

# now let’s define a function

A bald person wearing glasses

Description automatically generated with medium confidence

A group of men wearing sports uniforms

Description automatically generated with low confidence

A group of men

Description automatically generated with low confidence

Now we need to implement the same concept here:

Graphical user interface, application

Description automatically generated

We can’t write numbers as shown above:

Graphical user interface, application

Description automatically generated

Debug and see how it works:

def fact(n):

    if n == 0: # to set limit till where we need to go

        return 1 # we know that zero factorial is 1

    return n \* fact(n-1) # 5 \* fact(5-1) - calling the same function

result = fact(5)

print(result)

<https://www.programiz.com/python-programming/examples/km-mile>

kms = float(input('Enter the number of kilometres: '))

conv\_value = 0.62

miles = kms \* conv\_value

print(miles)

<https://www.programiz.com/python-programming/examples/random-number>

import random

rand\_int = random.randint(10)

print(rand\_int)

<https://www.programiz.com/python-programming/examples/swap-variables>

x = 10

y = 100

c = x

x = y

y = c

print('the value of x after swapping is: {}'.format(x))

print('the value of x after swapping is: {}'.format(y))