RECURSION

Recursion

Self Invocation-Circular Function

It is a programming technique in which a function calls itself either directly or indirectly

Recursion Vs Iteration

- Recursion and iteration both repeatedly executes the set of instructions
- The iteration is when a loop repeatedly executes until the controlling condition becomes false.
- Recursion is when a statement in a function calls itself repeatedly until the base case is reached.

```
def fun():
    d()
```

Function fun() is calling another function ()

```
def fun():
    fun()
```

Function fun() is calling itself from its own function body. It is a recursive function now.

Direct Recursion

```
def fun():
    d()
def  d():
    fun()
```

Function fun() is calling function d(), which is calling function fun()

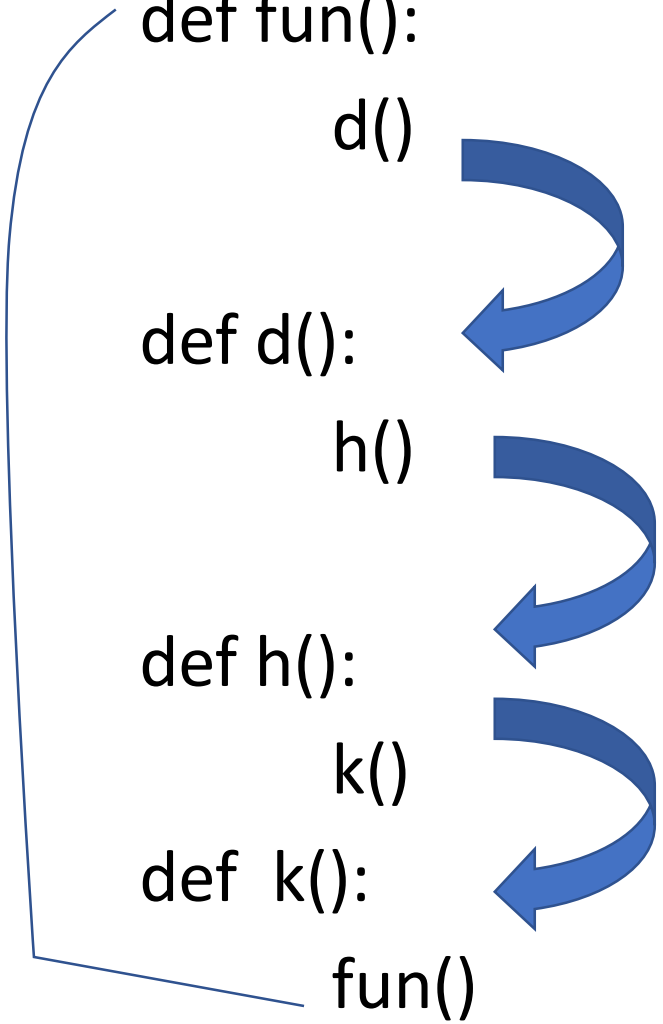
Indirect Recursion

Another Example of Indirect Recursion

```
def fun():
        d()

def d():
        h()

def h():
        k()
def  k():
        fun()
```

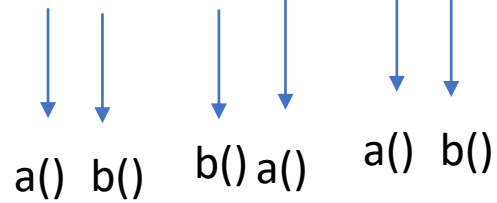


Flow of Execution in Recursion

```
1. def fun():
2.     print("hello")
3.     d()
4.  def  d():
5.      print("world")
6.      fun()
7. fun()
```

OUTPUT
hello
world
hello
world
hello
world
hello
world
.
.
.

7-1-2-3-4-5-6-**1-2-3-4-5-6-**1-2-3-4-5-6-**1-2-3-4-5-6**…….



fun() and d() will keep calling one another infinitely and the entire memory will used up

Mandatory requirements in Recursion Code

- Specify ending condition(clearly define **where to stop**).
- Base case- any condition where a **recursive function** does **not invoke /calls itself** .
- It **must have a base case**, whose **result is known to us**.
- **Base case** must be **reachable** for some arguments/parameter

Recursion Limit/depth

- Number of times the procedure is called recursively is called Recursion Limit/Depth.
- By default recursion limit is 1000.

1. To find the recursion limit:

```
import sys
print(sys.getrecursionlimit())
```

2. To set recursion limit

```
import sys
print(sys.setrecursionlimit(5000))
```

When infinite recursion occurs

1. Base case not defined
2. Base case not reached

Write a recursive function that finds sum of n numbers

```
def fun(n):
        if(n==1):
                return 1
        else:
                return(n+fun(n-1))

N=int(input(“enter no”))
D=fun(N)
print(D)
```

How recursion works:

```
1. def fun(n):
2.     if(n==1):
3.         return 1
4.     else:
5.         return(n+fun(n-1))
6. N=int(input("enter no"))
7. D=fun(N)
8. print(D)
```

7- D=fun(4)
1- def fun(4)
  5- return (4+fun(3))
1- def fun(3)
  5-return(3+fun(2))
1- def fun(2)
  5-return(2+fun(1))
1- def fun(1)
  3-return 1

Flow of execution

1-6-7-1-2-4-5-1-2-4-5-1-2-4-5-1-2-3-(loop moves until he reached the ending limit)-7-8

#Write a Python Program to determine whether a given number is even or odd recursively.

```
def check(n):
    if (n < 2):
        return (n % 2 == 0)
    return (check(n - 2))

n=int(input("Enter number:"))
if(check(n)==True):
    print("Number is even!")
else:
    print("Number is odd!")
```

```
'''
output
Enter number:4
Number is even!
'''
```

#Write a Python Program to find the fibonacci series using recursion.

```
def fibonacci(n):
    if(n <= 1):
        return n
    else:
        return(fibonacci(n-1) + fibonacci(n-2))

n = int(input("Enter number of terms:"))
print("Fibonacci sequence:")
for i in range(n):
    print(fibonacci(i), end=" ")
```

```
'''
output
Enter number of terms:5
Fibonacci sequence:
0 1 1 2 3
'''
```

#Write a Python Program to find the factorial of a number using recursion..

```
def factorial(n):
    if(n <= 1):
        return 1
    else:
        return(n*factorial(n-1))
n = int(input("Enter number:"))
print("Factorial:")
print(factorial(n))
```

```
'''
output
Enter number:5
Factorial:
120
'''
```

#Write recursive code to compute and print sum of squares of n numbers. Values of n is passed as parameter

```
def sqsum(n):
    if(n==1):
        return 1
    else:
        return n*n + sqsum(n-1)
n=int(input("enter value of n"))
print(sqsum(n))
```

```
'''
output
enter value of n3
14
'''
```

#Write a Python recursion function code to enter a number and find all the multiple of 3 from 1 till the input number.

```
def multi_3(n):
    if(n==1):
        return 3
    else:
        return (multi_3(n-1)+3)
n=int(input("enter a number"))
for i in range(1,n):
    print(multi_3(i))
```

```
'''
output
enter a number3
3
6
'''
```

Write a Python program to find the X^n using recursion function.

```
def x_power(x,n):
    if(n==0):
        return(1)
    else:
        return(x*x_power(x,n-1))

x=int(input("enter no"))
n=int(input("enter power"))
print(x_power(x,n))
```

```
'''
output
enter no3
enter power2
9
'''
```

#find output

```
def gcd(a,b):
    if(b==0):
        return a
    else:
        return (b, a%b)

print(gcd(20,12))
```

```
'''
output
(12,8)
'''
```