① Sum of first N Number

N = 3

o/p = 6

1+2+3 = 6

Parameter

functional

Parameterised

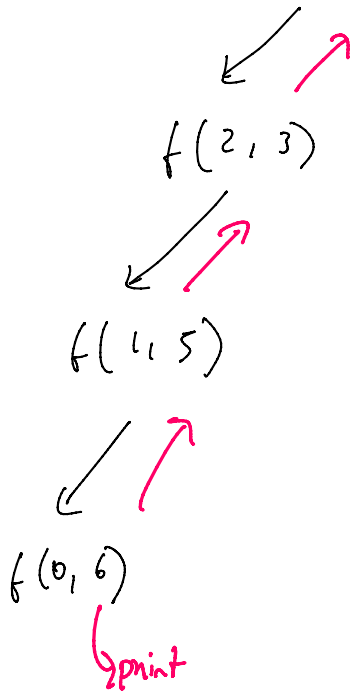
```
f(i, sum)
{
  if (i < 1)
  {
    print (sum)
    return
  }
  f(i-1, sum+i)
}

main()
{ n → 3
  f(n, 0)
}
```

f(3, 0) → f(2, 3) → f(1, 5) → f(0, 6)

f(0, 6): if (i < 1) ✓ → print (sum) 6 → return

Output : 6

f(3, 0)



Functional n = 3

3 + f(2) → 2 + f(1) → 1 + f(0) → 0

f(n) → Sum of first N nos

```
f(n)
{
  if (n == 0)
    return 0;

  return n + f(n-1);
}
main()
{
  n ← 3
```

3 + f(2) 3 + 3 = 6

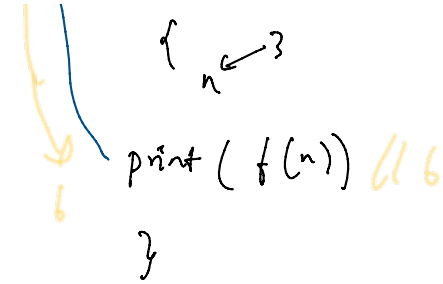
```
f(n)
{
  if (n == 0)
    return 0;

  return n + f(n-1);
}
```

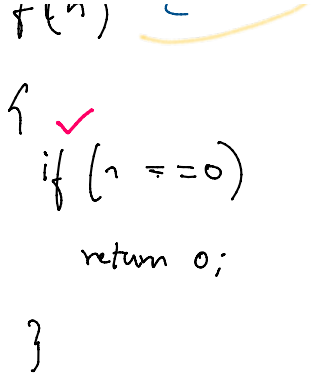
2 + f(1) 2 + 1 = 3

```
f(n)
{
  if (n == 0)
    return 0;

  return n + f(n-1);
}
```

1 + f(0) 1 + 0 = 1

```
f(n)
```





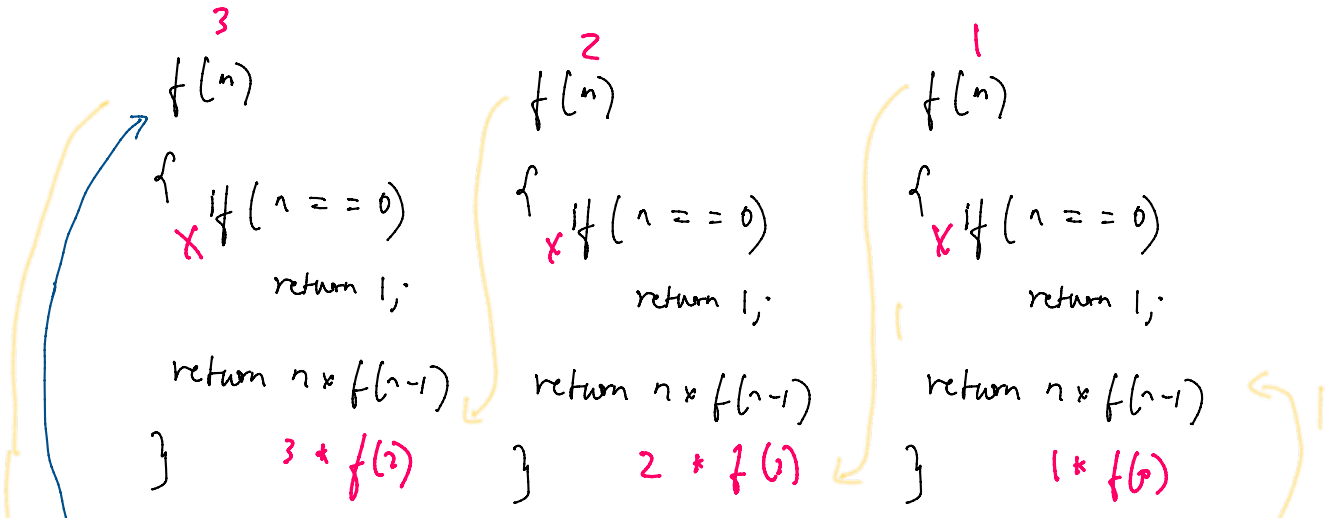

```cpp
#include<bits/stdc++.h>
int sum(int n){

    // base condition
    if(n == 0)
        return 0;
    // recursive call
    return n + sum(n - 1);
}

int main(){
using namespace std;
    int n = 5;
    cout<<sum(n);
    return 0;
}
```

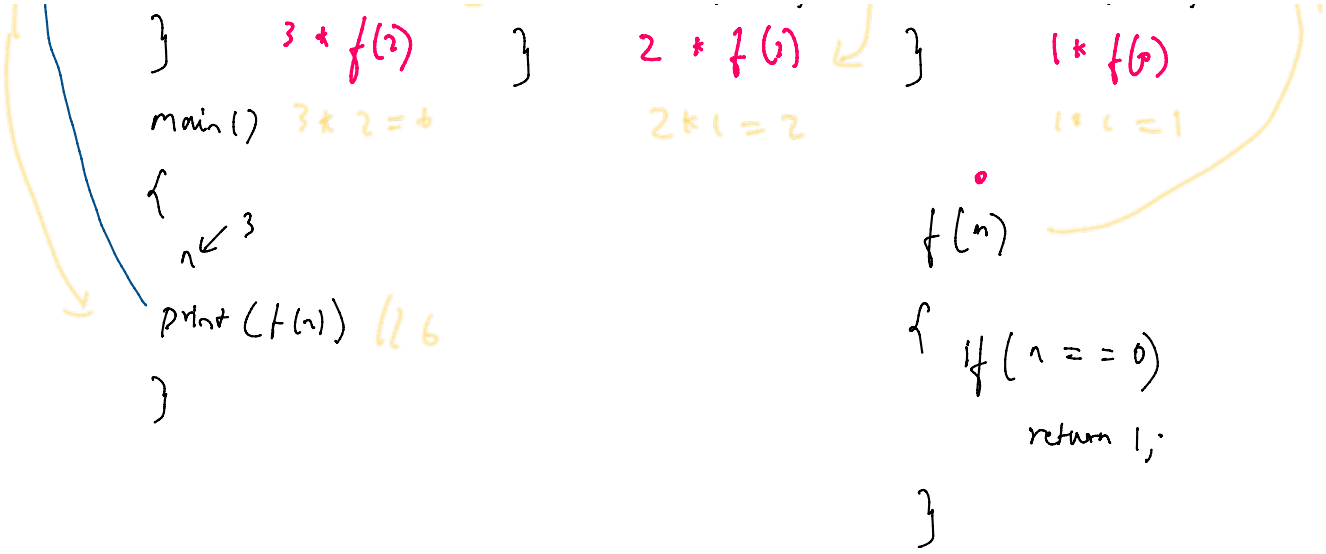
```
1 1

outputf.in
1 15

ished in 2.6s]
```

(2) Factorial of N

N = 2 ⇒ 2 × 1 = 2

N = 3 ⇒ 3 × 2 × 1 = 6






```cpp
#include<bits/stdc++.h>
int fac(int n){

    // base condition
    if(n == 0)
        return 1;
    // recursive call
    return n * fac(n - 1);
}

int main(){
using namespace std;
    int n = 3;
    cout<<fac(n);
    return 0;
}
```

```
1 1

outputf.in
1 6

nished in 2.0s]
```

T.C ⇒ O(N) S.C ⇒ O(N) (Stack Space)