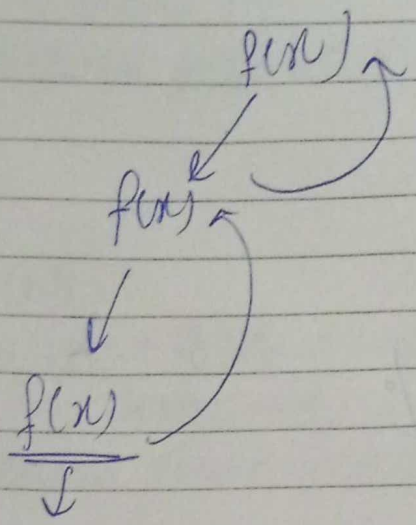
Recursion

Date __/__/____

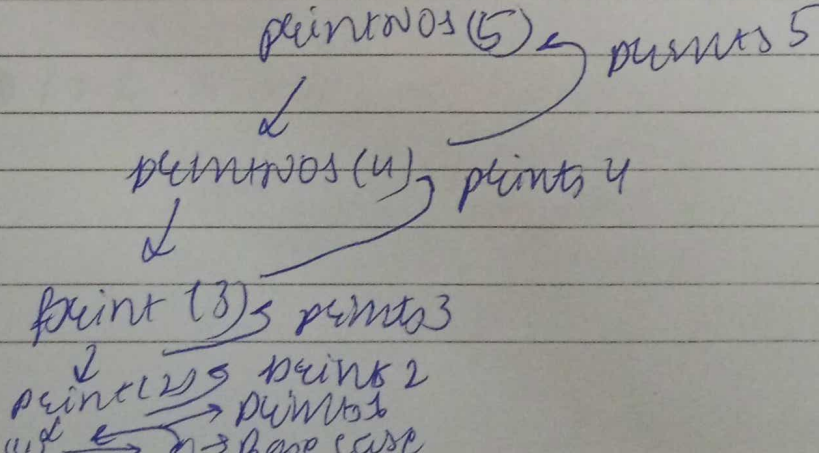


Basic condⁿ satisfied
like till N != 0

Print 1 - N

```
void printnos(N){
    if (N==0){
        return;
    }
    printnos(N-1);
    cout<<N<<" ";
}
```

1 2 3 4 5

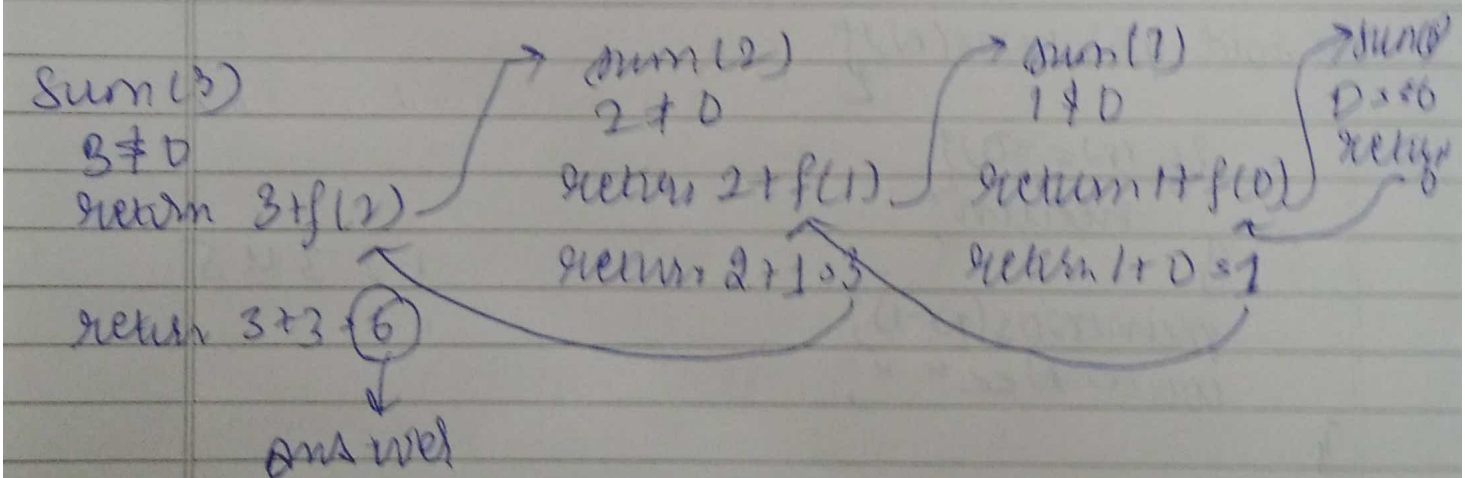


Date / /

Q Sum of first N nos.

```
int sum(N){
    sum = 0
    if(N==0){
        return sum;
    }
    return sum + f(n-1);
}
```

```
int sum(N){
    if(N==0){
        return 0;
    return n + f(n-1);
}
```



Date.__/__/____

① fact(N)
if (N == 0 || 1)
return 1;
return N * fact(N-1)

② fib(N)
if (n == 0 or n == 1)
return 1;
return fib(n-1) + fib(N-2)

O(2^n) time complexity
because
n → n-1, n-2
n-1 → n-2, n-3
n-2 → n-3, n-4

fib(5) → 8 → Answer
↓
return fib(4) + fib(3)
↓ ↑ 5
return fib(3) + fib(2) ← 2
↓ ↑ 3
return fib(2) + fib(1) = 1 + 2 = 3
↓ ↑ 2
fib(1) fib(0)

return fib(2) + fib(1) = 2
↓ ↑ 2
fib(1) + fib(0) ← 1
↓ 1
1 + 1 = 2

1 1 2 3 5 8
f(n) = 3 + 2
= f(n-1) + f(n-2)

Date __/__/____

Reverse an array

```
f(l, r)
    if (l ≥ r) {
        return;
    swap(a[l], a[r]);
    f(l+1, r-1);
```

```
f(i)
    if (i >= n/2)
        return;
    swap(a[i], a[n-i-1])
    f(i+1);
```

Palindrome string

```
f(s)
    if (s.length() <= 1) return 1;
    if (s[0] == s[s.length()-1]) {
        f(s.substr(1, s.length()-2));
    else
        return 0;
```

f(MADAM) → f(ADA) → f(D)

f(MADAM) → M == M → f(ADA)

f(ADA) → A == A → f(D)

f(D) → length <= 1 → return 1

1 → answer = 1 or true

just like array

```
f(i)
  if (i>=n/2) return true;
  if (s[i] != s[n-i-1])
        return false;
  return f(i+1);
```