Date: / /
Page:

Day - 73

Recursion - 2

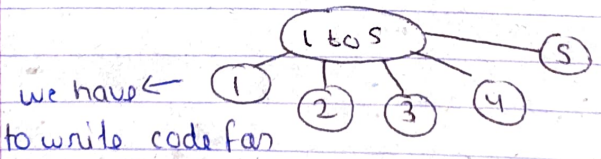
* Print 1 to N

⇒ Iterative Approach

```
for( i=1; i<=N; i++)
    cout<<i;
```

⇒ Recursive approach

```
void Print(int num, int N){
    if(num==N){
        cout << num;
        return;
    }
    cout << num <<endl;
    Print(num+1, N);
}
int main(){
    Print(1, 5);
}
```



we have ← to write code for this only & other will handle automatically.

⇒ Mathematical Function:

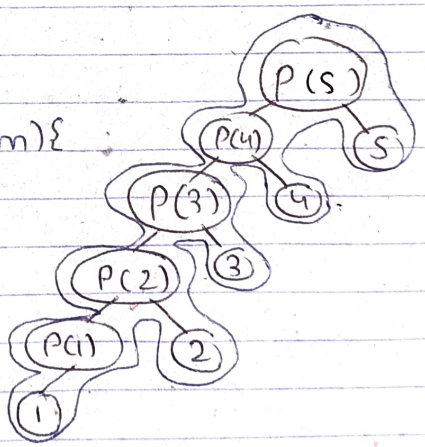
Print(5, N) = 5

Print(4, N) = 4, Print(5, N)

Print(3, N) = 3, Print (4, N)
Print(2, N) = 2, Print (3, N)
Print(1, N) = 1, Print(2, N)

Print(num, N) = num, Print(num+1, N)

⇒ So if we want to do this by using only one argument then ——
First we start from num & goes to 1 —
5
4
3
2
1 → after reaching one we will start printing
1 2 3 4 5



⇒
```
void Print( int num){
    if( num == 1 ){
        cout << 1 ;
        return
    }
    Print( num - 1 );
    cout << num;
}
```

⇒ Print(1) = 1
Print(2) = Print(1), 2
P(3) = P(2), 3
P(4) = P(3), 4
P(5) = P(4), 5
P(N) = P(N-1), N

Date: / /
Page:

⇒ Print 1 to N (Even no)

```
printeven (int num, int N){
    if(num > N)
        return;
    cout << num;
    printeven (num+2, N);
}

printeven2 ( int num){
    if( num == 2 ){
        cout << 2 <<endl;
        return;
    printeven2 ( num-2);
    cout <<num <<endl;
}
```

⇒

P(2) = 2
P(4) = P(2), 4
P(6) = P(4), 6
⋮
P(n) = P(n-2), n