Session 09
Monday, 27 October 2025 9:24 AM

Recursion :- function call itself.
→ Directly
→ Indirectly

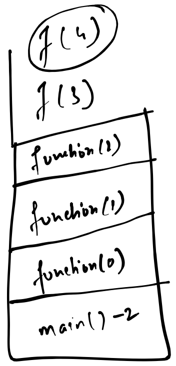
```
function ( int i )
{ -①
② printf(" %d ", i );
③ function( i+1 );
④ }
```

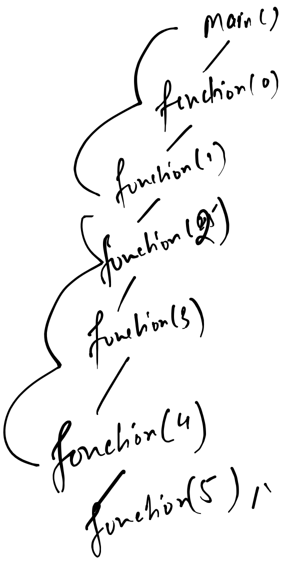
if(i >= 5)
return;

main()
① → 3
② → function(0);
③ → 5

0
1
2

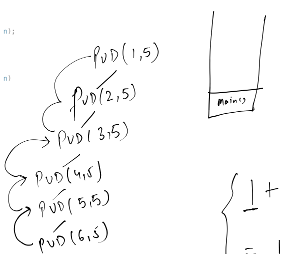
Last in first out
Stack





(Recursion Tree)

```
#include <stdio.h>
void printuserDefine(int i, int n);
int main()
{
    printuserDefine(1, 5);
    return 0;
}
void printuserDefine(int i, int n)
{
    if (i > n)
        return;
    printuserDefine(i + 1, n);
    printf("%d \n", i);
}
```



5
4
3
2
1

1 + 2 + 3 + 4 + 5
= 1 + sum(2,5);

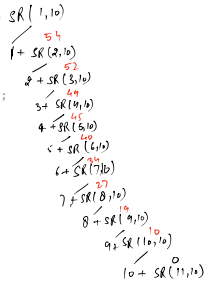
= 5 + sum(4)
4 + sum(3)
3 + sum(2)
2 + sum(1);
1

function which adds first n numbers.

```
int Sum ( int n )
{
    if( n == 1 )
        return 1;

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

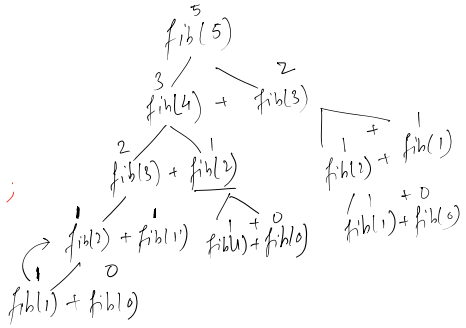
```
int sumRange(int start, int end)
{
    if (start > end)
        return 0;
    return start + sumRange(start + 1, end);
}
```




```
int fib(int n)
{
    if (n <= 1)
        return n;

    return fib(n-1) + fib(n-2);
}
```

0 1
0 1




```
void reverseString(char* s, int sSize) {
}
```

char temp;
while (i<j)
temp = s[i];
s[i] = s[j];
s[j] = temp;
i++;
j--;