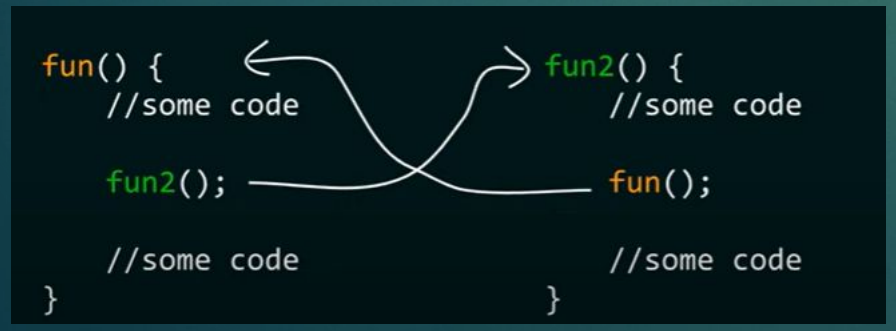
RECURSION IN C

BY: DR. SHEFALI ARORA CHOUHAN

RECURSIVE FUNCTION

- A recursive function is a function that calls itself again
- Activation record is maintained on a stack
- Return address of call & local variables are maintained

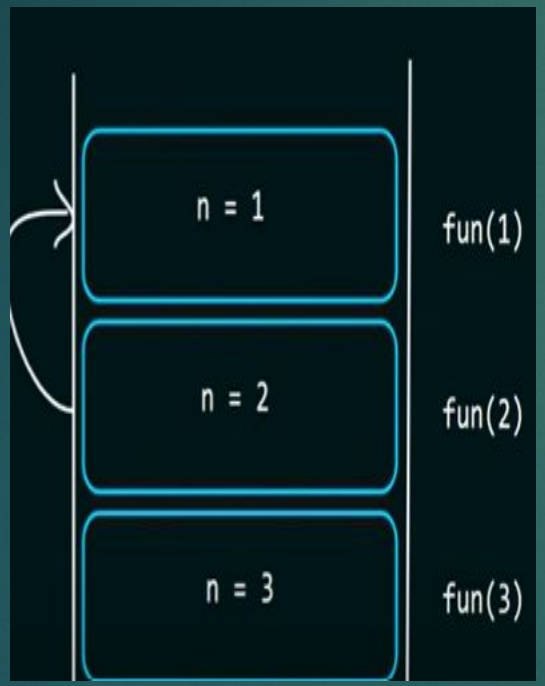
```
int fun()
{
    ...
    fun();
}
```



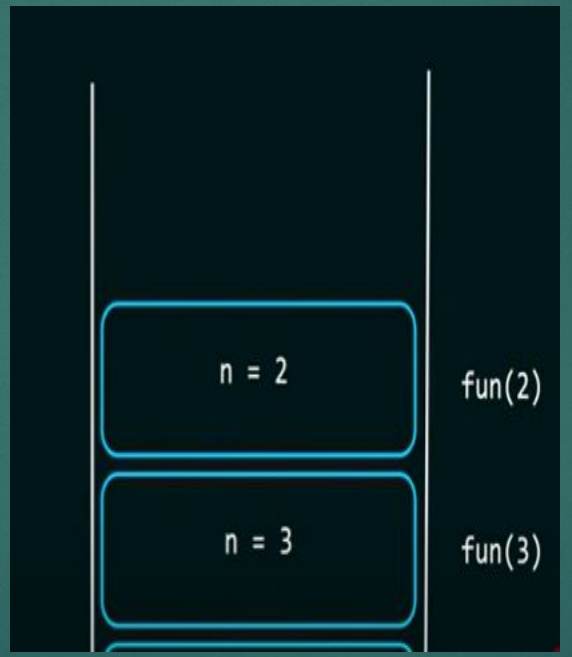

```
int fun(int n)
{
    if( n==1 )
        return 1;
    else
        return 1 + fun( n-1 );
}

int main() {
    int n = 3;
    printf("%d", fun(n));
    return 0;
```

USING STACKS: THE PUSH & POP CONCEPT



1



1+1

1+1+1

CALCULATE FACTORIAL USING RECURSION

```
Fact( )
{
      if(    )
      {
          ...
      }
      else
      {
          ...
      }
```

Base Case

Recursive Procedure

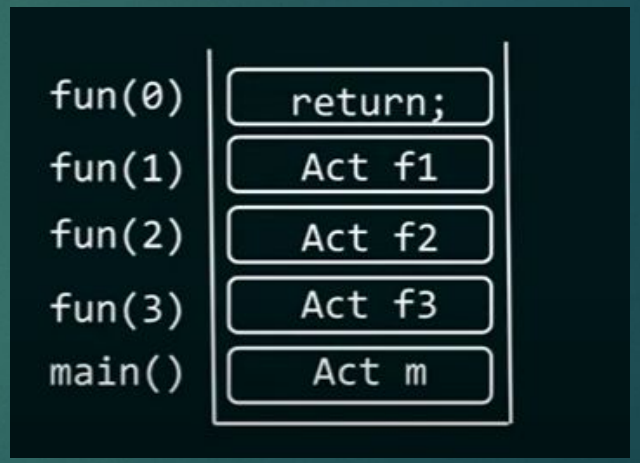
```
Fact(int n)
{
    if( n == 1)
    {
        return 1;
    }
    else
    {
        return n * Fact(n-1);
    }

}
```

HEAD VS TAIL RECURSION

- A function is tail recursive if recursion is the last thing done by the function in a program

```
void fun(int n) {
    if(n == 0)
        return;
    else
        printf("%d ", n);
    return fun(n-1);
}
int main() {
    fun(3);
    return 0;
```



HEAD VS TAIL RECURSION

- A function is non-tail recursive if there is something left to evaluate

```
void fun(int n) {
    if(n == 0)
        return;
    fun(n-1);
    printf("%d ", n);
}
int main() {
    fun(3);
    return 0;
```