Recursion
Step-by-Step

Recursive Function Execution

```
int fact(int N) {
    if (N <= 1)          //line 1
        return 1;        //line 2
    return N*fact(N-1); //line 3
}
```

```
int main(){
     ….
     … fact(5) …
     …
}
```

```
int fact(int N=5) {
    if (N <= 1)          //line 1
        return 1;        //line 2
    return 5*fact(5-1); //line 3
}
```

N*fact(N-1)

```
int fact(int N=4) {
    if (N <= 1)           //line 1
        return 1;         //line 2
    return 4*fact(4-1); //line 3
}
```

```
int fact(int N=5) {
    if (N <= 1)           //line 1
        return 1;         //line 2
    return 5*fact(5-1); //line 3
}
```

```
int fact(int N=3) {
    if (N <= 1)          //line 1
        return 1;        //line 2
    return 3*fact(3-1); //line 3
}
```

```
int fact(int N=4) {
    if (N <= 1)          //line 1
        return 1;        //line 2
    return 4*fact(4-1); //line 3
}
```

```
int fact(int N=5) {
    if (N <= 1)          //line 1
        return 1;        //line 2
    return 5*fact(5-1); //line 3
}
```

```
int fact(int N=2) {
    if (N <= 1)         //line 1
        return 1;       //line 2
    return 2*fact(2-1); //line 3
}
```

```
int fact(int N=3) {
    if (N <= 1)         //line 1
        return 1;       //line 2
    return 3*fact(3-1); //line 3
}
```

```
int fact(int N=4) {
    if (N <= 1)         //line 1
        return 1;       //line 2
    return 4*fact(4-1); //line 3
}
```

```
int fact(int N=5) {
    if (N <= 1)         //line 1
        return 1;       //line 2
    return 5*fact(5-1); //line 3
}
```

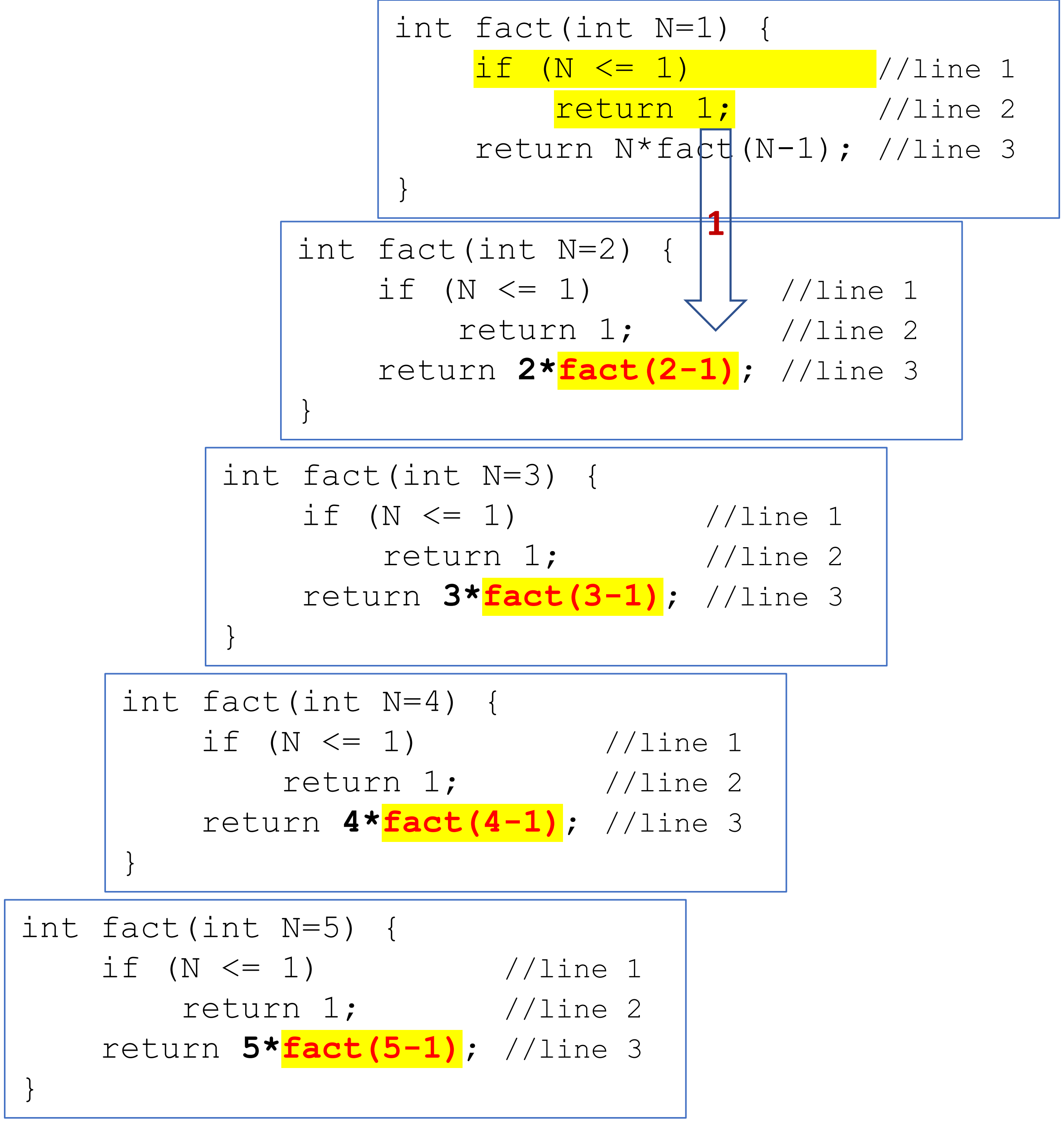
```
int fact(int N=1) {
    if (N <= 1)             //line 1
        return 1;           //line 2
    return N*fact(N-1);     //line 3
}
```

```
int fact(int N=2) {
    if (N <= 1)             //line 1
        return 1;           //line 2
    return 2*fact(2-1);     //line 3
}
```

```
int fact(int N=3) {
    if (N <= 1)             //line 1
        return 1;           //line 2
    return 3*fact(3-1);     //line 3
}
```

```
int fact(int N=4) {
    if (N <= 1)             //line 1
        return 1;           //line 2
    return 4*fact(4-1);     //line 3
}
```

```
int fact(int N=5) {
    if (N <= 1)             //line 1
        return 1;           //line 2
    return 5*fact(5-1);     //line 3
}
```



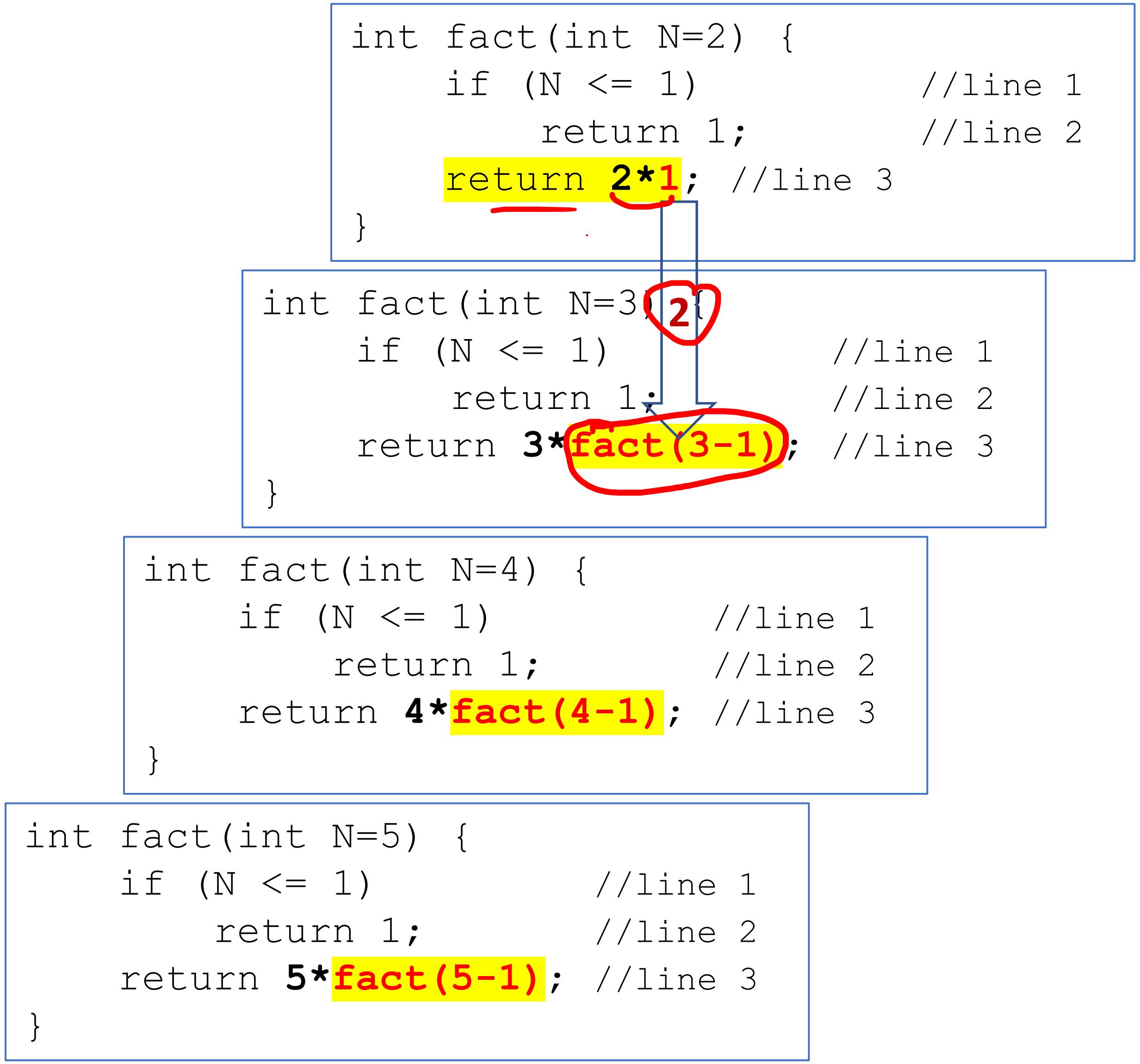
```
int fact(int N=1) {
    if (N <= 1)          //line 1
        return 1;        //line 2
    return N*fact(N-1);  //line 3
}
```

```
int fact(int N=2) {
    if (N <= 1)          //line 1
        return 1;        //line 2
    return 2*fact(2-1);  //line 3
}
```

```
int fact(int N=3) {
    if (N <= 1)          //line 1
        return 1;        //line 2
    return 3*fact(3-1);  //line 3
}
```

```
int fact(int N=4) {
    if (N <= 1)          //line 1
        return 1;        //line 2
    return 4*fact(4-1);  //line 3
}
```

```
int fact(int N=5) {
    if (N <= 1)          //line 1
        return 1;        //line 2
    return 5*fact(5-1);  //line 3
}
```

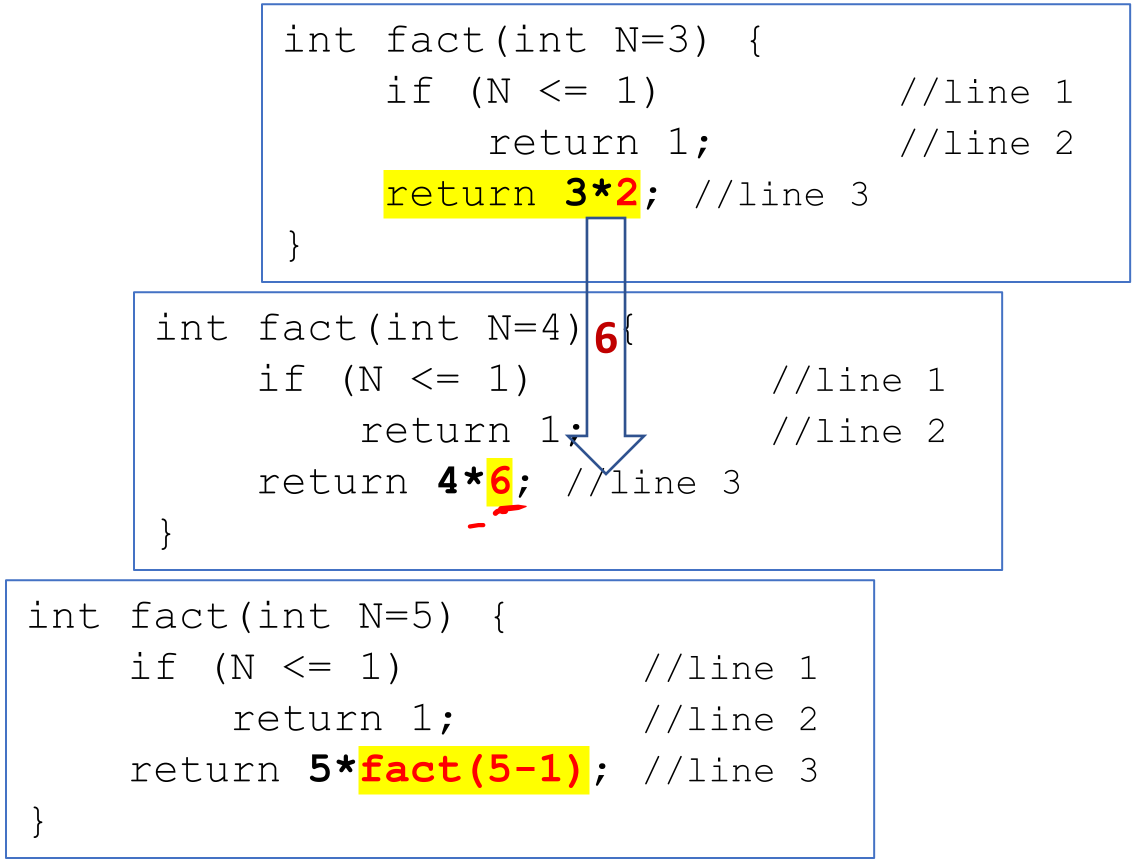


```
int fact(int N=2) {
    if (N <= 1)             //line 1
        return 1;           //line 2
    return 2*1; //line 3
}
```

```
int fact(int N=3) {
    if (N <= 1)             //line 1
        return 1;           //line 2
    return 3*fact(3-1); //line 3
}
```

```
int fact(int N=4) {
    if (N <= 1)             //line 1
        return 1;           //line 2
    return 4*fact(4-1); //line 3
}
```

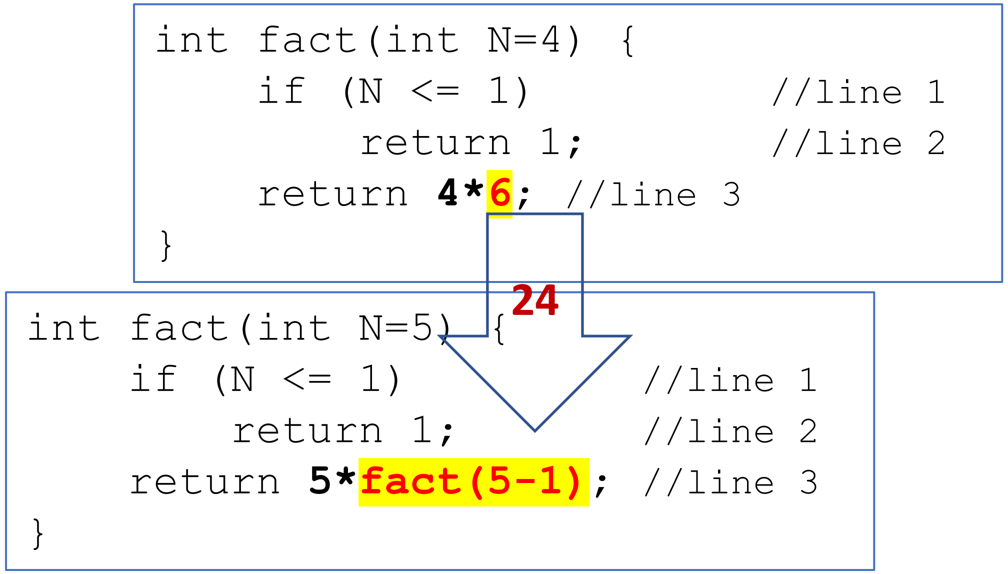
```
int fact(int N=5) {
    if (N <= 1)             //line 1
        return 1;           //line 2
    return 5*fact(5-1); //line 3
}
```



```
int fact(int N=3) {
    if (N <= 1)            //line 1
        return 1;          //line 2
    return 3*2; //line 3
}
```

```
int fact(int N=4) {
    if (N <= 1)            //line 1
        return 1;          //line 2
    return 4*6; //line 3
}
```

```
int fact(int N=5) {
    if (N <= 1)            //line 1
        return 1;          //line 2
    return 5*fact(5-1); //line 3
}
```



```
int fact(int N=4) {
    if (N <= 1)         //line 1
        return 1;       //line 2
    return 4*6; //line 3
}
```

```
int fact(int N=5) {
    if (N <= 1)         //line 1
        return 1;       //line 2
    return 5*fact(5-1); //line 3
}
```

```
int fact(int N=5) {
    if (N <= 1)          //line 1
        return 1;        //line 2
    return 5*24; //line 3
}
```

```
int main(){
     ….
     …  120…
     …
}
```

Recursive Function Execution

Stack

Heap

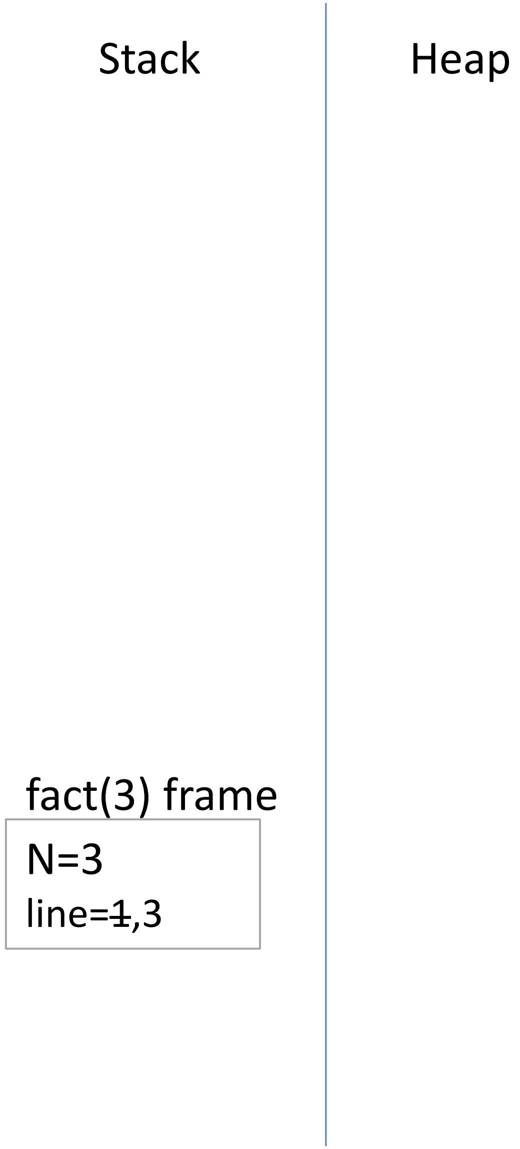
```
int fact(int N) {
    if (N <= 1)          //line 1
        return 1;        //line 2
    return N*fact(N-1); //line 3
}
```

```
int main(){
... fact(3) … }
```

Recursive Function Execution

```
int fact(int N) {
    if (N <= 1)           //line 1
        return 1;         //line 2
    return N*fact(N-1); //line 3
}
```

```
int main(){
... fact(3) … }
```



Stack

Heap

fact(3) frame

N=3

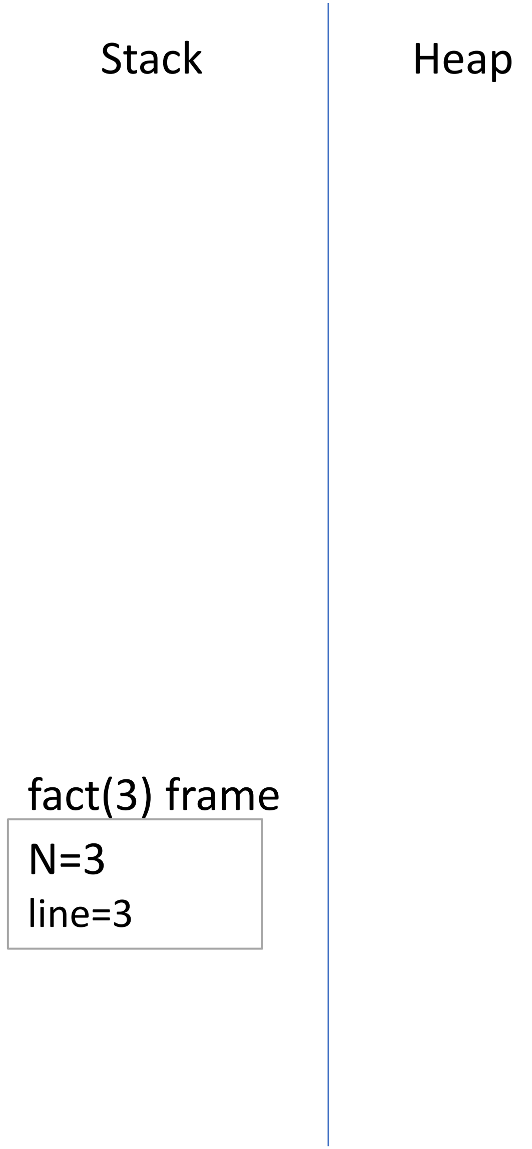
line=~~1~~,3

Recursive Function Execution

```
int fact(int N) {
    if (N <= 1)           //line 1
        return 1;         //line 2
    return N*fact(N-1);   //line 3
}
```

```
int fact(int N=3) {
    if (N <= 1) return 1;
    return 3*fact(2);
}
```

```
int main(){
... fact(3) … }
```



Stack

Heap

fact(3) frame

N=3

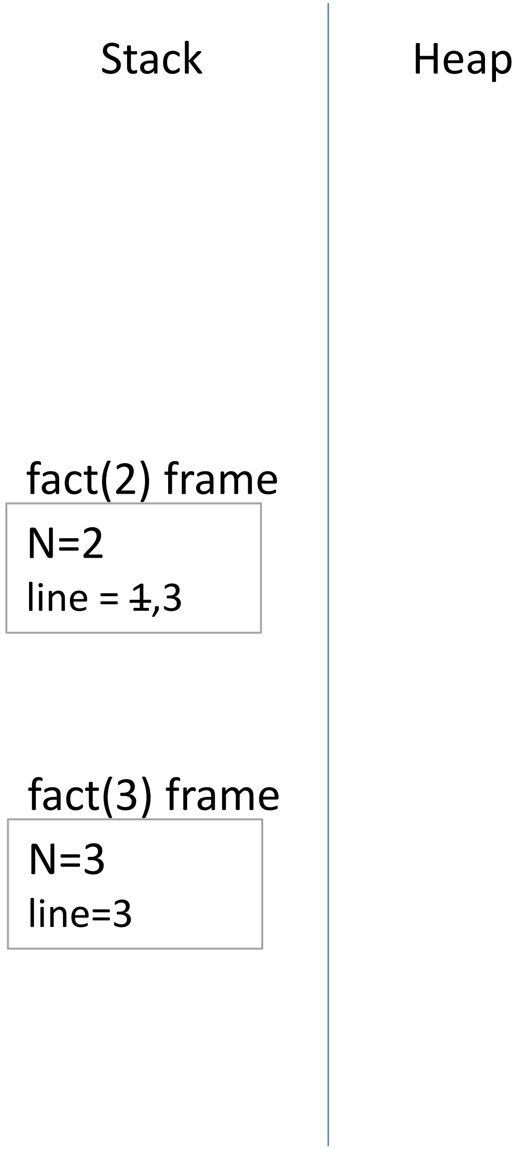
line=3

Recursive Function Execution

```
int fact(int N) {
    if (N <= 1)          //line 1
        return 1;        //line 2
    return N*fact(N-1); //line 3
}
```

```
int fact(int N=3) {
    if (N <= 1) return 1;
    return 3*fact(2);
}
```

```
int main(){
... fact(3) … }
```



Stack

Heap

fact(2) frame

N=2

line = ~~1~~,3

fact(3) frame

N=3

line=3

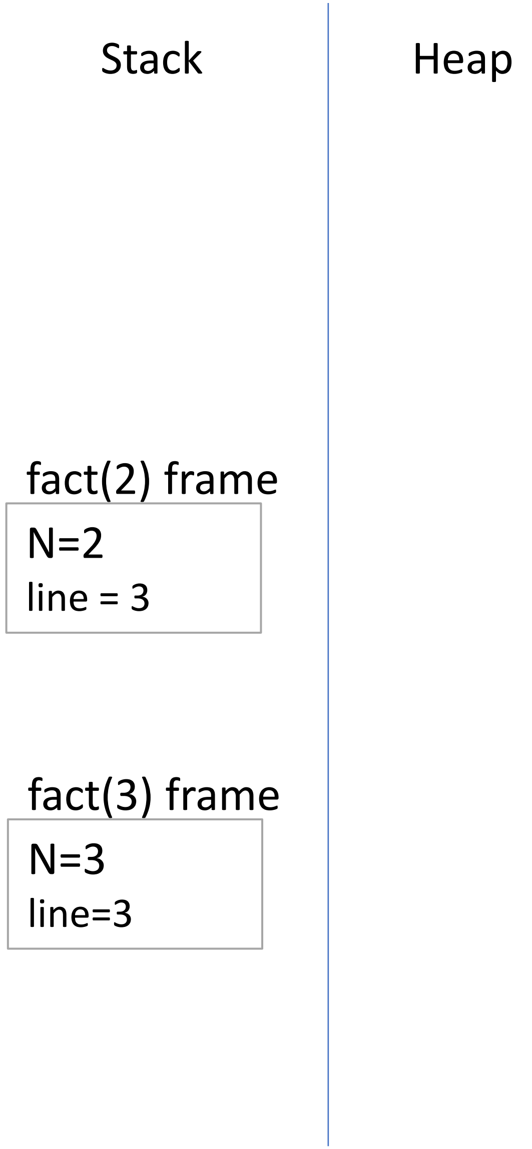
Recursive Function Execution

```
int fact(int N) {
    if (N <= 1)           //line 1
        return 1;         //line 2
    return N*fact(N-1); //line 3
}
```

```
int fact(int N=2) {
    if (N <= 1) return 1;
    return 2*fact(1);
}
```

```
int fact(int N=3) {
    if (N <= 1) return 1;
    return 3*fact(2);
}
```

```
int main(){
... fact(3) … }
```



Stack

Heap

fact(2) frame

N=2
line = 3

fact(3) frame

N=3
line=3

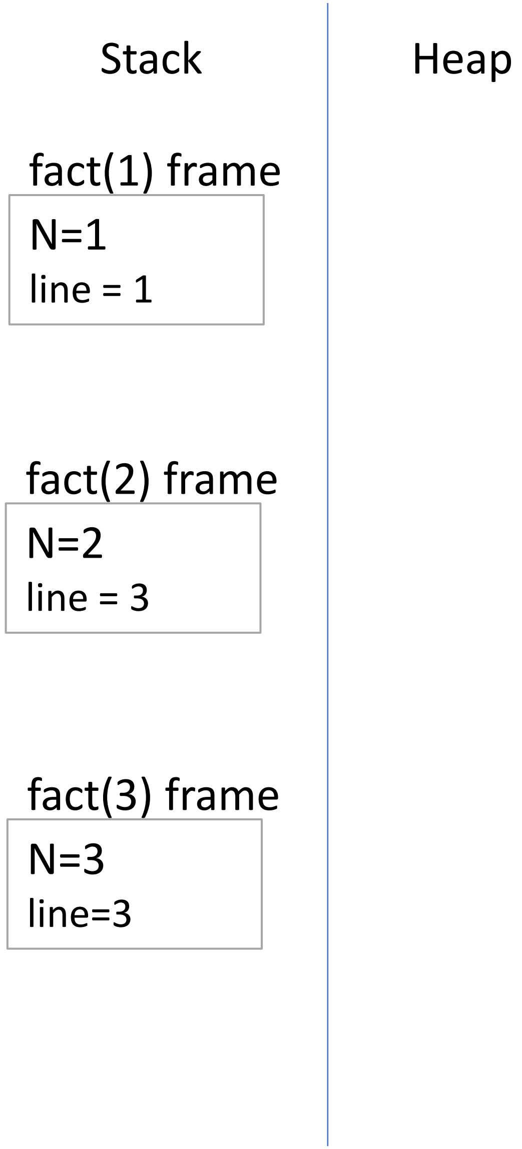
Recursive Function Execution

```
int fact(int N) {
    if (N <= 1)          //line 1
        return 1;        //line 2
    return N*fact(N-1); //line 3
}
```

```
int fact(int N=2) {
    if (N <= 1) return 1;
    return 2*fact(1);
}
```

```
int fact(int N=3) {
    if (N <= 1) return 1;
    return 3*fact(2);
}
```

```
int main(){
... fact(3) … }
```



Recursive Function Execution

```
int fact(int N) {
    if (N <= 1)          //line 1
        return 1;        //line 2
    return N*fact(N-1); //line 3
}
```

```
int fact(int N=1) {
    if (N <= 1) return 1;
    return N*fact(N-1);
}
```

```
int fact(int N=2) {
    if (N <= 1) return 1;
    return 2*fact(1);
}
```

```
int fact(int N=3) {
    if (N <= 1) return 1;
    return 3*fact(2);
}
```

```
int main(){
... fact(3) … }
```

Stack

fact(1) frame

N=1
line = 1

fact(2) frame

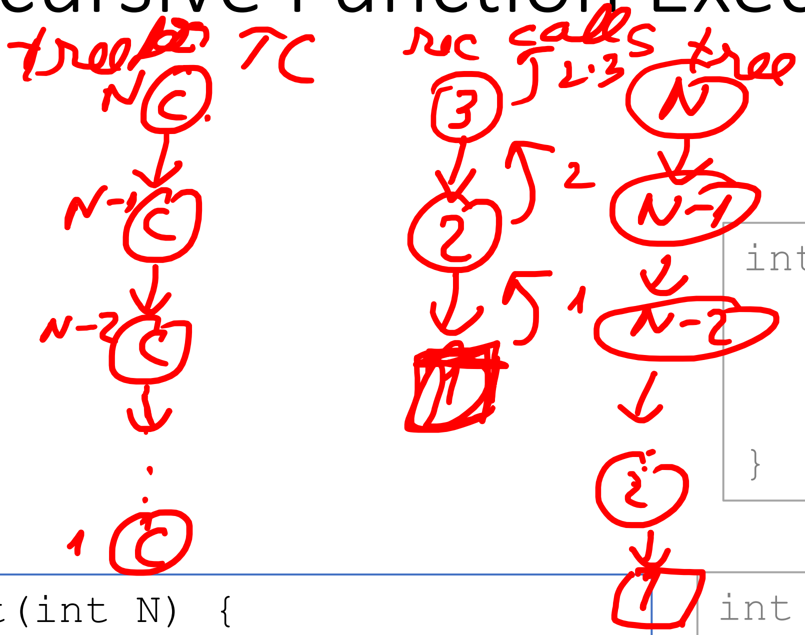
N=2
line = 3

fact(3) frame

N=3
line=3

Heap

Recursive Function Execution



```
int fact(int N) {
    if (N <= 1)        C      //line 1
        return 1;             //line 2
    return N*fact(N-1);       //line 3
}
```

$\underset{fact(N)}{TC} = \underbrace{C + C + \dots + C}_{N\text{ nodes}}$

N times

```
int fact(int N=1) {
    if (N <= 1)
        return 1;
    return N*fact(N-1);
}
```

1

```
int fact(int N=2) {
    if (N <= 1) return 1;
    return 2*fact(1);
}
```

```
int fact(int N=3) {
    if (N <= 1) return 1;
    return 3*fact(2);
}
```

```
int main(){
... fact(3) … }
```

Stack

fact(1) frame

N=1
line = 2

fact(2) frame

N=2
line = 3

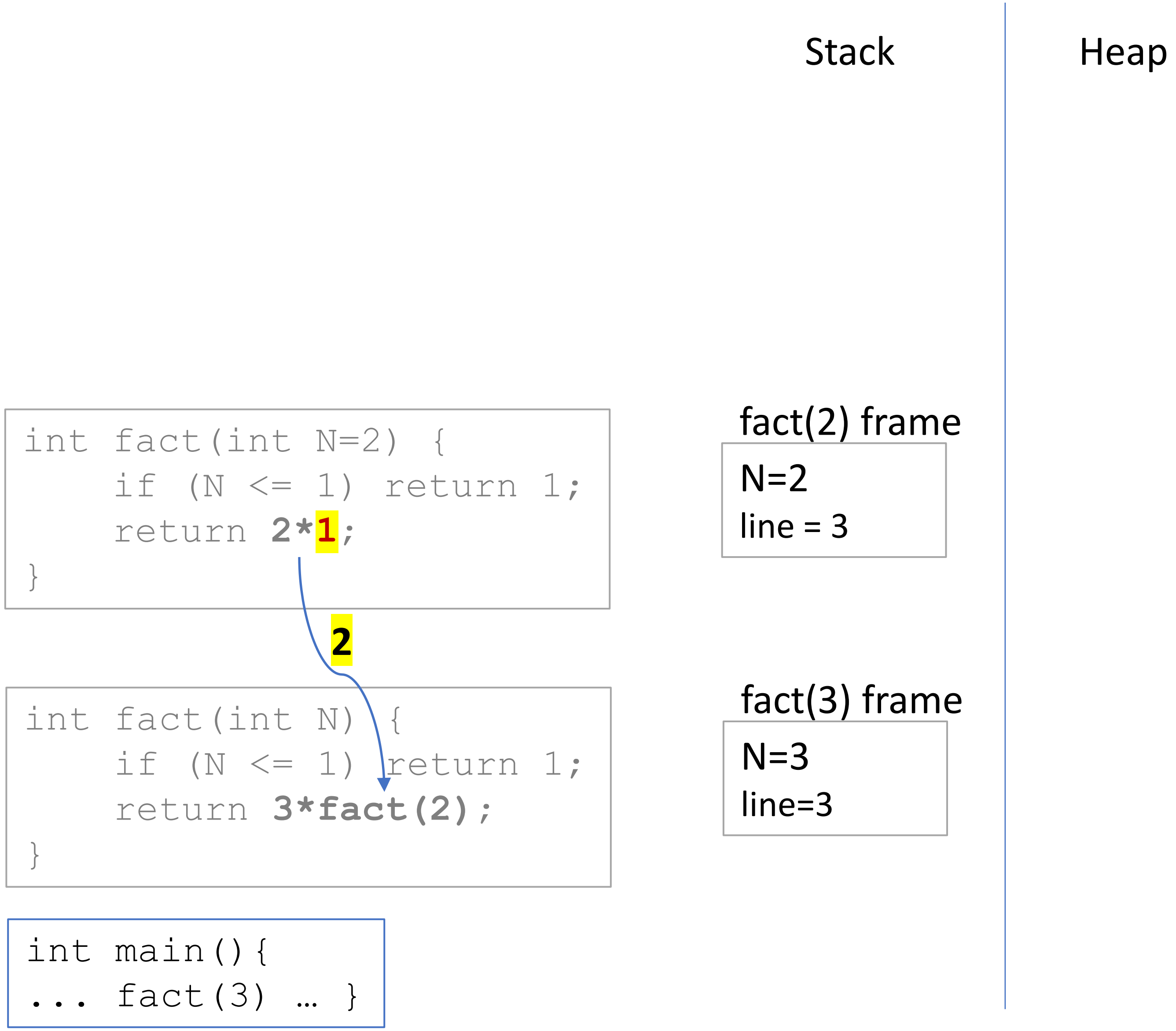
fact(3) frame

N=3
line=3

Heap

Recursive Function Execution

```
int fact(int N) {
    if (N <= 1)          //line 1
        return 1;        //line 2
    return N*fact(N-1); //line 3
}
```



Stack

Heap

```
int fact(int N=2) {
    if (N <= 1) return 1;
    return 2*1;
}
```

```
int fact(int N) {
    if (N <= 1) return 1;
    return 3*fact(2);
}
```

```
int main(){
... fact(3) … }
```

fact(2) frame

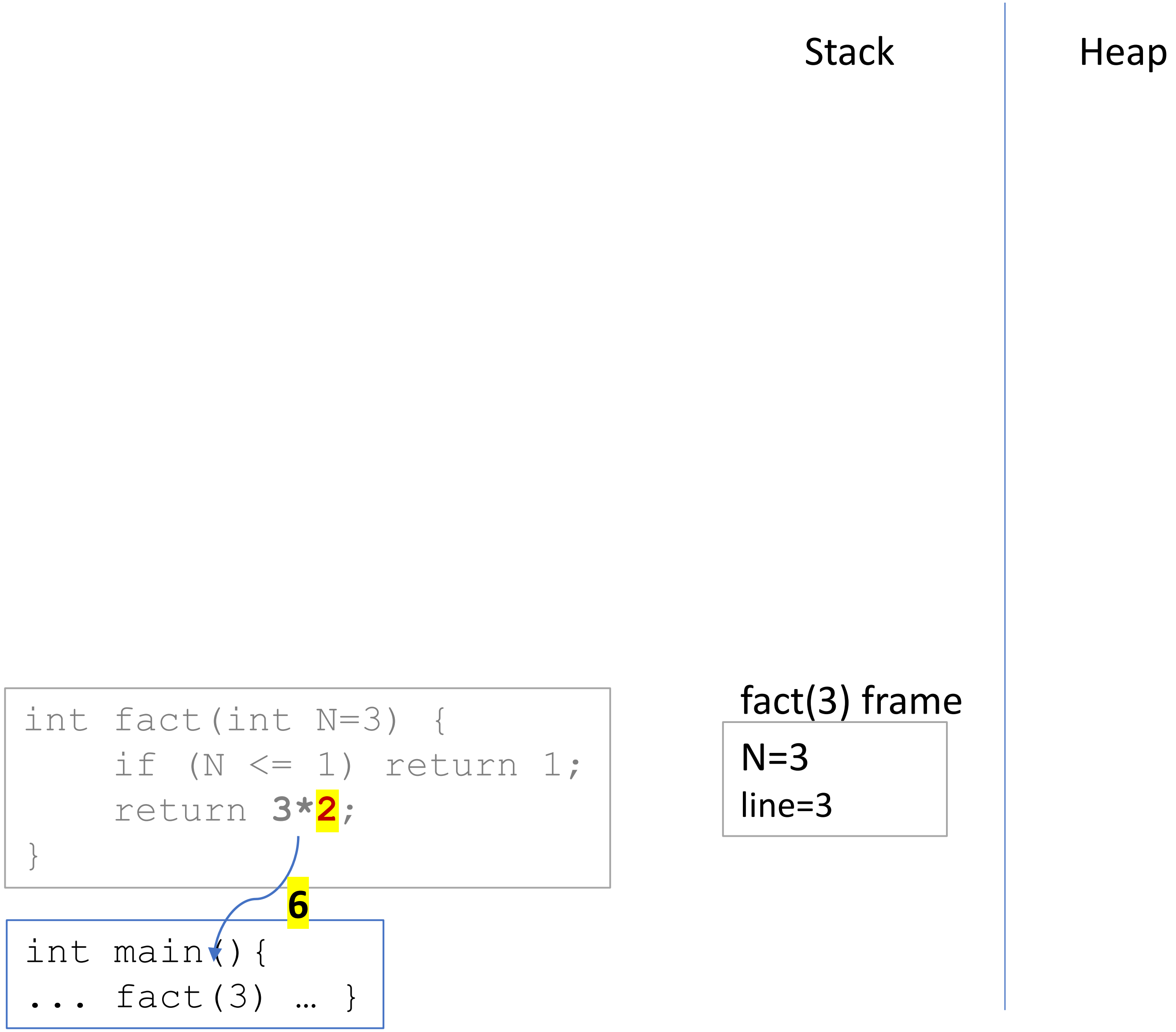
N=2
line = 3

fact(3) frame

N=3
line=3

Recursive Function Execution

```
int fact(int N) {
    if (N <= 1)           //line 1
        return 1;         //line 2
    return N*fact(N-1);  //line 3
}
```



Stack

Heap

```
int fact(int N=3) {
    if (N <= 1) return 1;
    return 3*2;
}
```

6

```
int main(){
... fact(3) … }
```

fact(3) frame

N=3
line=3

Recursive Function Execution

Stack

Heap

```
int fact(int N) {
    if (N <= 1)          //line 1
        return 1;        //line 2
    return N*fact(N-1); //line 3
}
```

```
int main(){
... 6 … }
```