

# Functions & Recursion (1A)

---

Copyright (c) 2010 - 2016 Young W. Lim.

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.2 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the section entitled "GNU Free Documentation License".

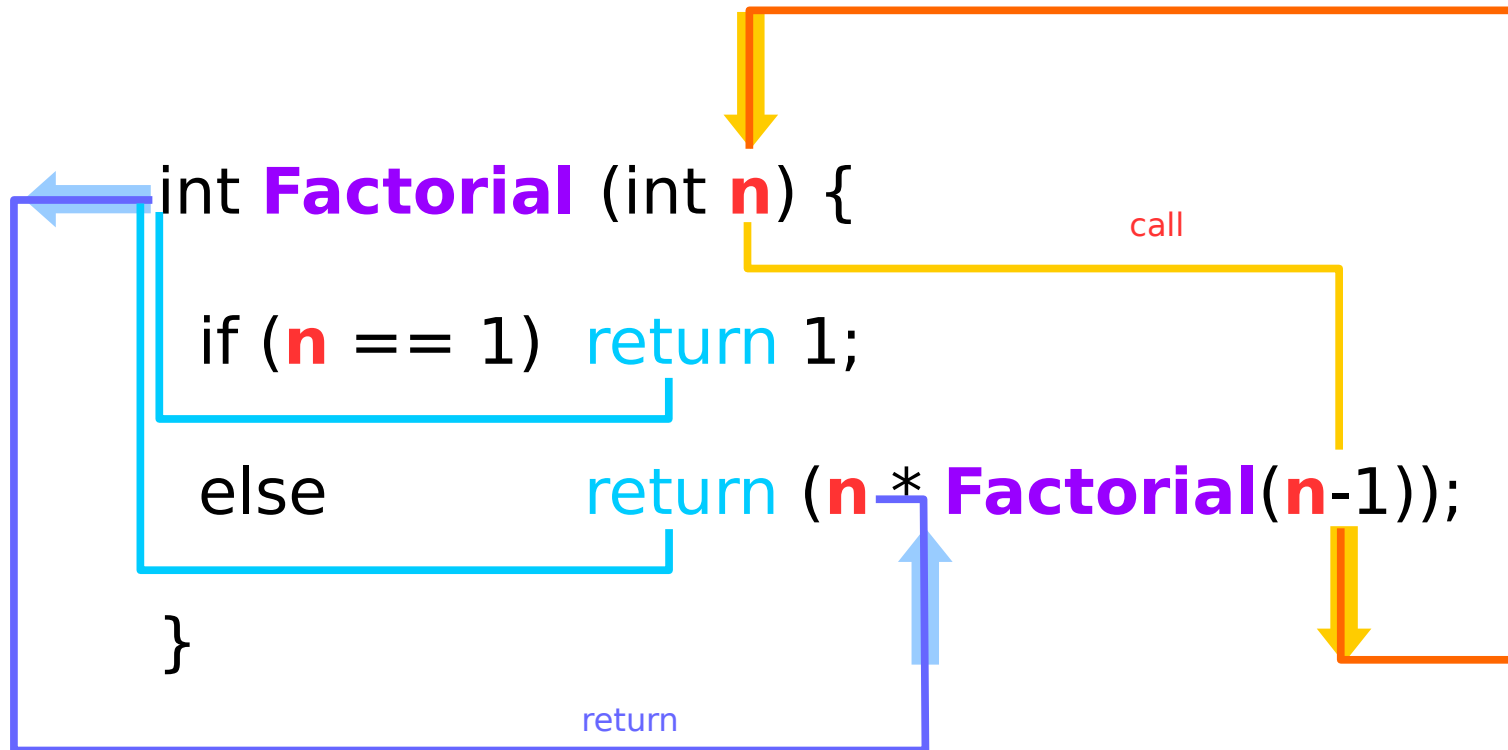
Please send corrections (or suggestions) to [youngwlim@hotmail.com](mailto:youngwlim@hotmail.com).

This document was produced by using OpenOffice.

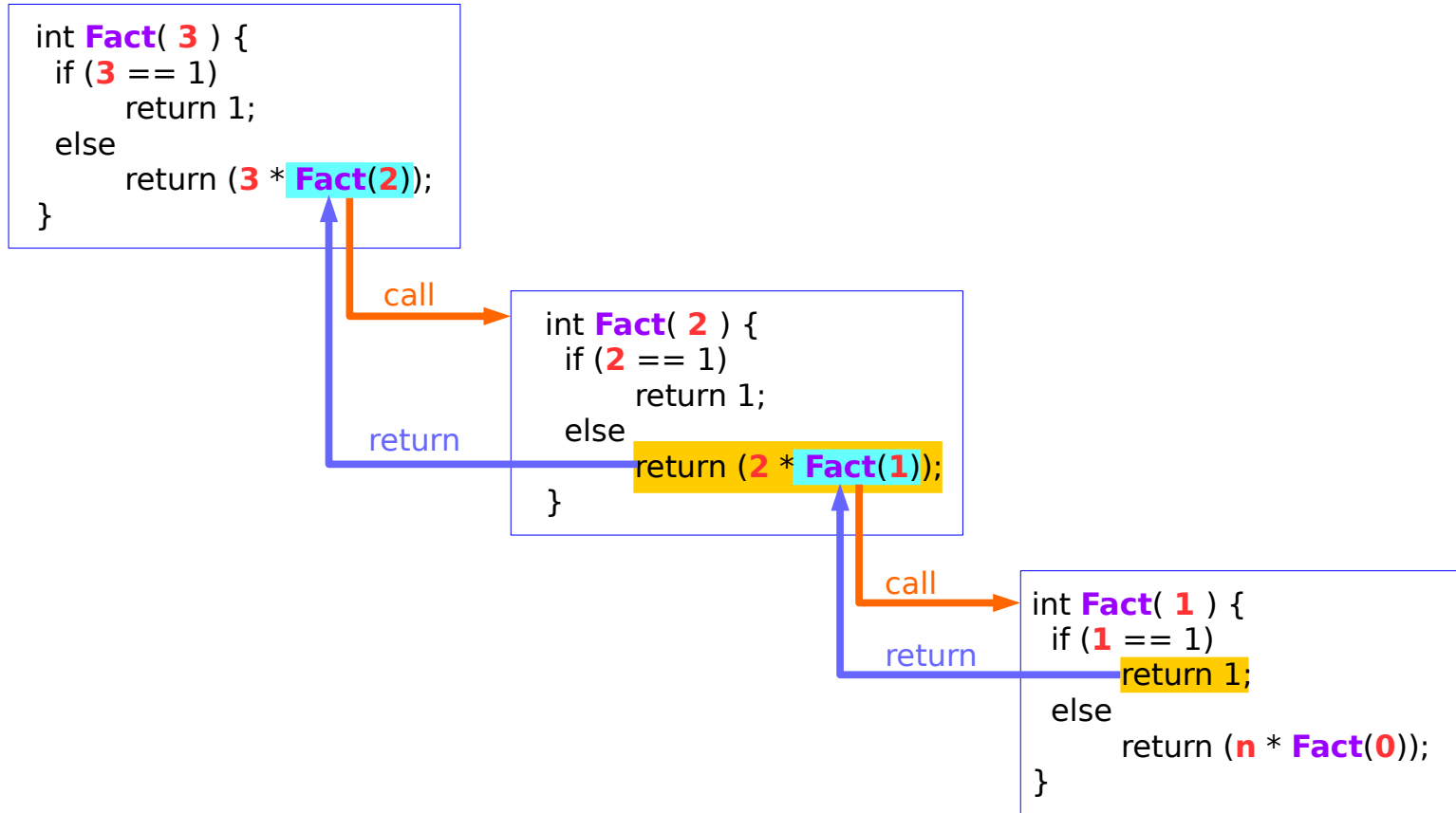
# Iterative Factorial Function

```
int Factorial (int n) {  
    int i, P=1;  
    for (i=n; i>0; i--) P *= i;  
    return (P);  
}
```

# Recursive Factorial Function



# Unrolled Function Calls and Returns

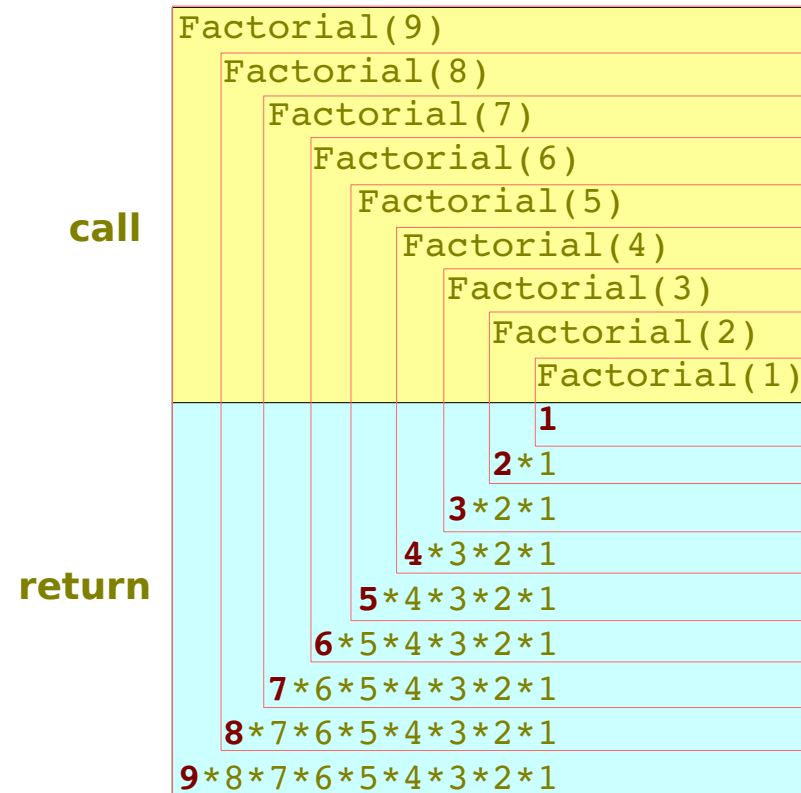


# Calling and Returning Sequence

```
#include <stdio.h>
```

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

```
int main(void) {  
    int f = Factorial( 9 );  
  
    printf("Factorial(9) = %d \n", f);  
}
```



# Verifying

```
#include <stdio.h>
```

```
int Factorial(int n) {  
    int tmp;
```

```
    printf("n=%d ", n);  
    if (n == 1) printf("Fact(1) = 1 \n");  
    else printf("Fact(%d) = %d*Fact(%d)\n", n, n, n-1);
```

```
    if (n == 1) return 1;  
    else {
```

```
        tmp = Factorial(n-1);  
        printf("==> Fact(%d) = %d\n", n-1, tmp);  
        return (n*tmp);  
    }
```

```
    }
```

```
int main(void) {
```

```
    Factorial( 9 );
```

```
}
```

```
n=9 Fact(9) = 9*Fact(8)  
n=8 Fact(8) = 8*Fact(7)  
n=7 Fact(7) = 7*Fact(6)  
n=6 Fact(6) = 6*Fact(5)  
n=5 Fact(5) = 5*Fact(4)  
n=4 Fact(4) = 4*Fact(3)  
n=3 Fact(3) = 3*Fact(2)  
n=2 Fact(2) = 2*Fact(1)  
n=1 Fact(1) = 1
```

```
==> Fact(1) = 1  
==> Fact(2) = 2  
==> Fact(3) = 6  
==> Fact(4) = 24  
==> Fact(5) = 120  
==> Fact(6) = 720  
==> Fact(7) = 5040  
==> Fact(8) = 40320  
==> Fact(9) = 362880
```

## References

- [1] Essential C, Nick Parlante
- [2] Efficient C Programming, Mark A. Weiss
- [3] C A Reference Manual, Samuel P. Harbison & Guy L. Steele Jr.
- [4] C Language Express, I. K. Chun