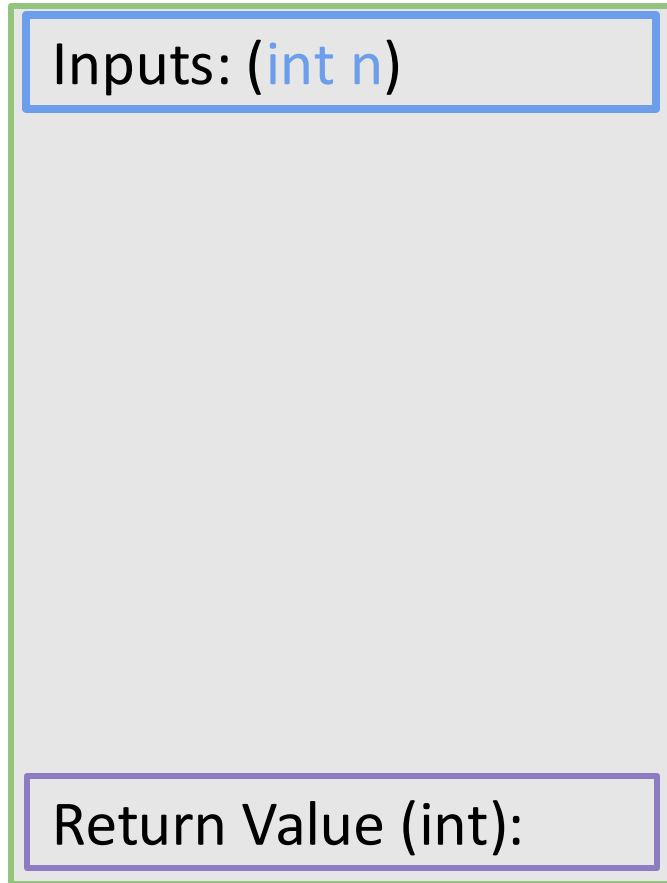


ECS 36A, May 23, 2025

Review of Recursion

- Recursion is tricky the first time you see it
 - And the second
 - And the third
 - ...
- You *will* get used to it
- Here's a review using Fibonacci numbers
- Slides done by Karen Tu, a TA for an earlier ECS 36A class (thanks, Karen!), modified for ECS 36A Spring 2025

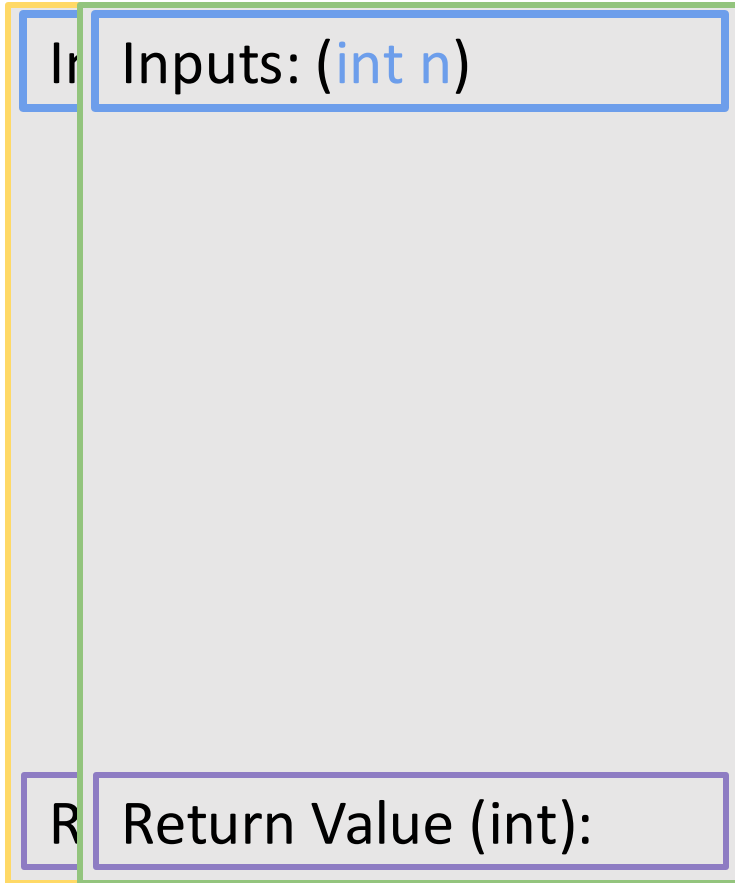
Function Demonstration Notations



An area in memory allocated for a function call.

int int n

Function Demonstration Notations



Yellow outline means the function space is allocated but the program is not running in that allocated space.

Green outline means the program is currently executing in this function space.

Example Function Call: Recursion

```
/* function declaration */
int fib(int n)
{
    int a, b;

    if (n <= 1)
        return n;
    a = fib(n-1);
    b = fib(n-2);
    return a + b;
}

int main(void)
{
    printf("%d\n", fib(4));
    return 0;
}
```

Inputs: (int n) 4

Return Value (int):

Example Function Call: Recursion

```
/* function declaration */
int fib(int n) n ← 4
{
    int a, b;

    if (n <= 1)
        return n;
    a = fib(n-1);
    b = fib(n-2);
    return a + b;
}

int main(void)
{
    printf("%d\n", fib(4));
    return 0;
}
```

Inputs: (int n) 4

int a

int b

Return Value (int):

Example Function Call: Recursion

```
/* function declaration */
int fib(int n) n ← 4
{
    int a, b;

    if (n <= 1) n = 4 > 1
        return n;
    a = fib(n-1);
    b = fib(n-2);
    return a + b;
}

int main(void)
{
    printf("%d\n", fib(4));
    return 0;
}
```

Inputs: (int n) 4

int a

int b

Return Value (int):

Example Function Call: Recursion

```
/* function declaration */
int fib(int n) n ← 4
{
    int a, b;

    if (n <= 1) n > 1
        return n;
    a = fib(n-1); a ← fib(4-1) = fib(3)
    b = fib(n-2);
    return a + b;
}

int main(void)
{
    printf("%d\n", fib(4));
    return 0;
}
```

Inputs: (int n) 3

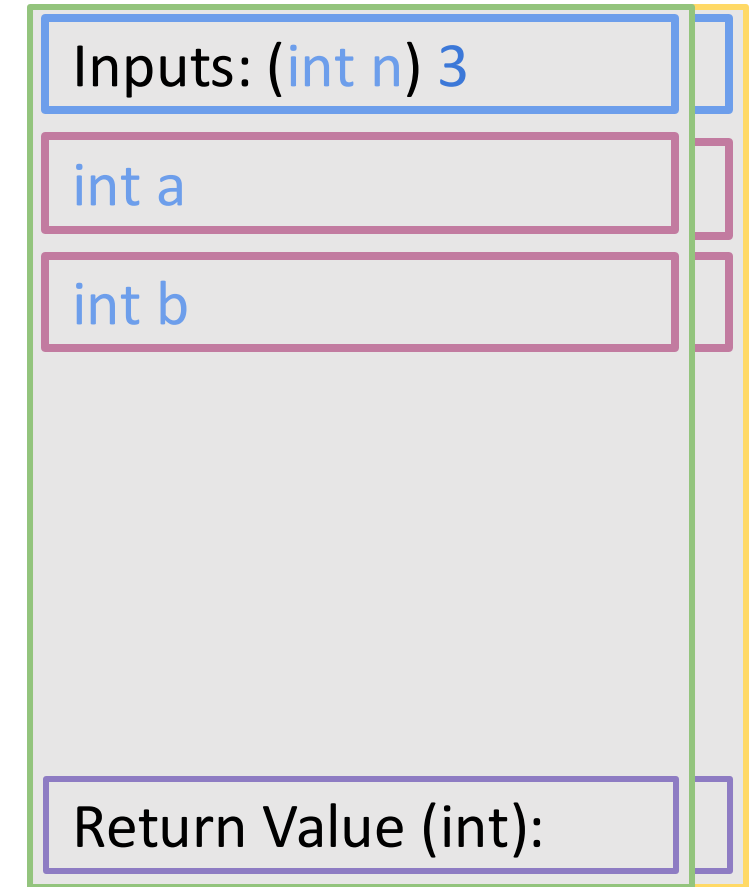
Return Value (int):

Example Function Call: Recursion

```
/* function declaration */
int fib(int n) n ← 3
{
    int a, b;

    if (n <= 1)
        return n;
    a = fib(n-1);
    b = fib(n-2);
    return a + b;
}

int main(void)
{
    printf("%d\n", fib(4));
    return 0;
}
```

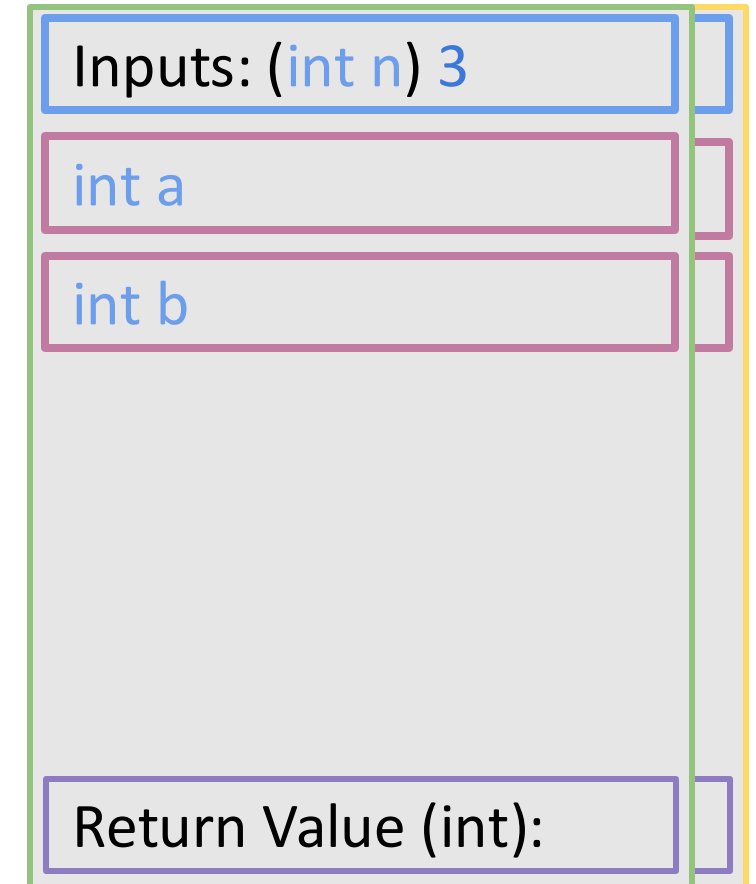


Example Function Call: Recursion

```
/* function declaration */
int fib(int n) n ← 3
{
    int a, b;

    if (n <= 1) n = 3 > 1
        return n;
    a = fib(n-1); a ← fib(3-1) = fib(2)
    b = fib(n-2);
    return a + b;
}

int main(void)
{
    printf("%d\n", fib(4));
    return 0;
}
```

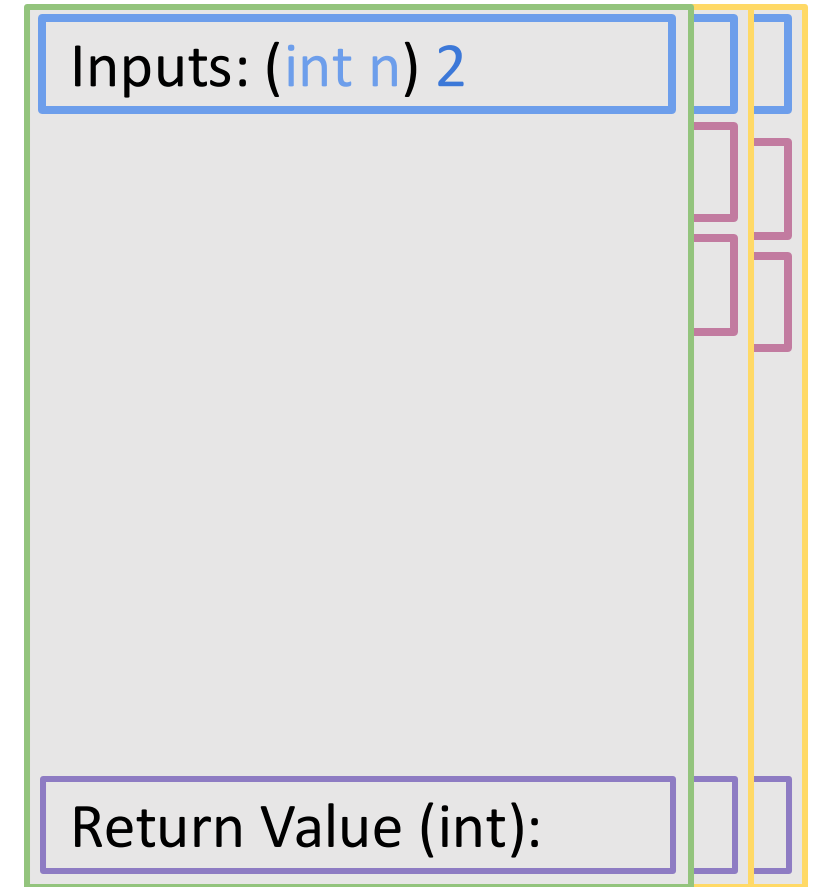


Example Function Call: Recursion

```
/* function declaration */
int fib(int n) n ← 2
{
    int a, b;

    if (n <= 1)
        return n;
    a = fib(n-1);
    b = fib(n-2);
    return a + b;
}

int main(void)
{
    printf("%d\n", fib(4));
    return 0;
}
```

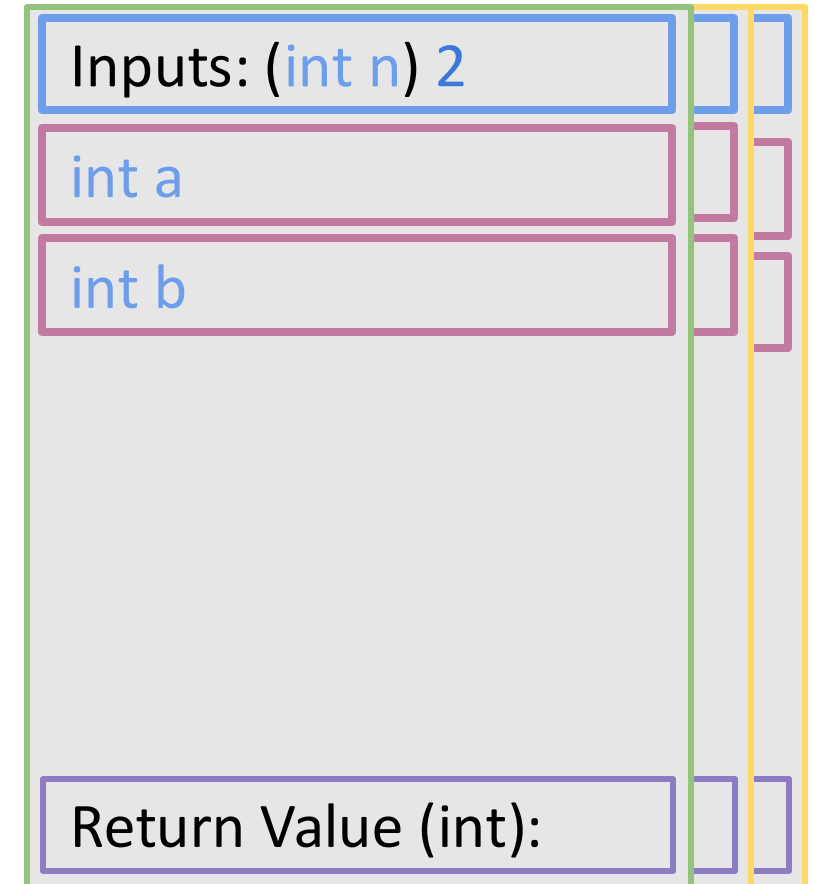


Example Function Call: Recursion

```
/* function declaration */
int fib(int n) n ← 2
{
    int a, b;

    if (n <= 1) n = 2 > 1
        return n;
    a = fib(n-1);
    b = fib(n-2);
    return a + b;
}

int main(void)
{
    printf("%d\n", fib(4));
    return 0;
}
```

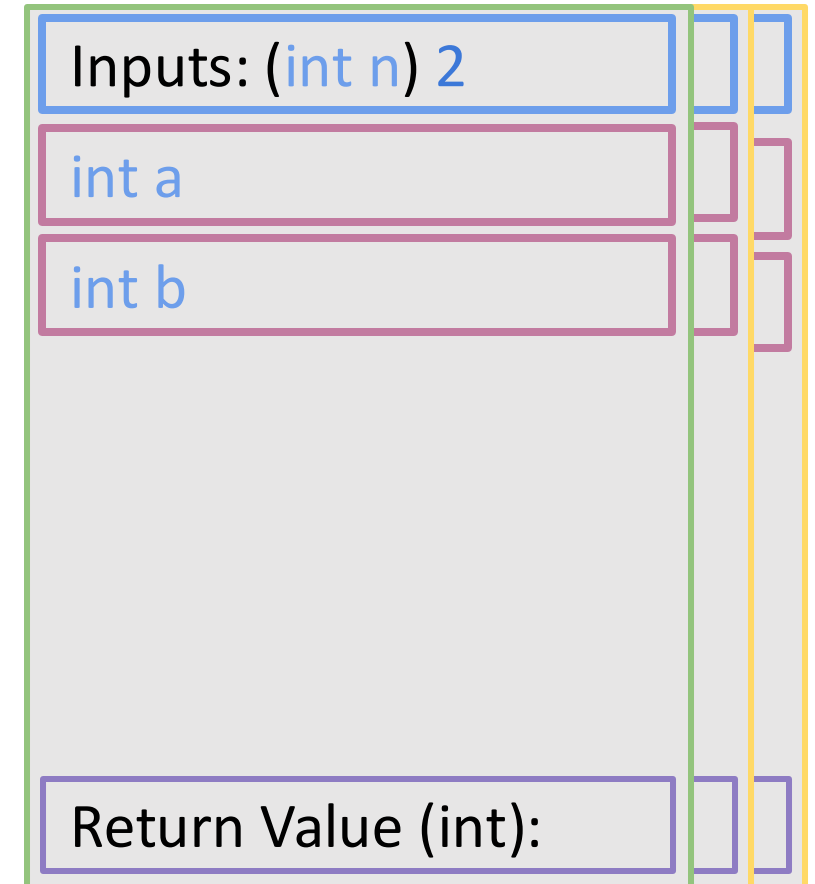


Example Function Call: Recursion

```
/* function declaration */
int fib(int n) n ← 2
{
    int a, b;

    if (n <= 1) n = 2 > 1
        return n;
    a = fib(n-1); a ← fib(2-1) = fib(1)
    b = fib(n-2);
    return a + b;
}

int main(void)
{
    printf("%d\n", fib(4));
    return 0;
}
```

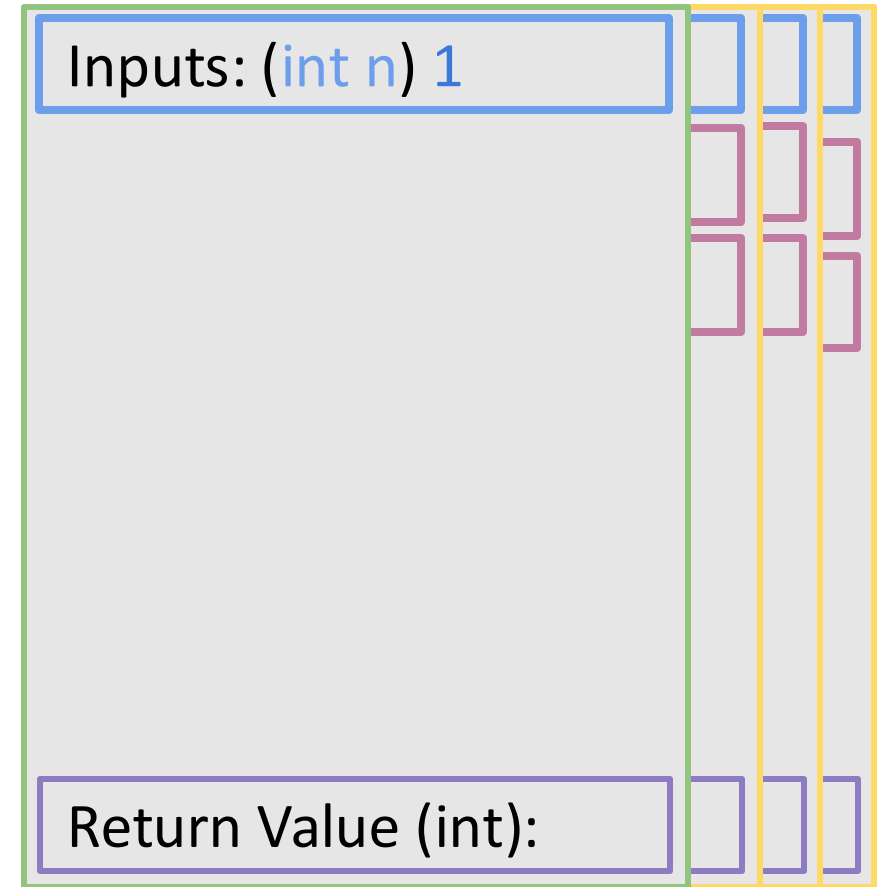


Example Function Call: Recursion

```
/* function declaration */
int fib(int n) n ← 1
{
    int a, b;

    if (n <= 1)
        return n;
    a = fib(n-1);
    b = fib(n-2);
    return a + b;
}

int main(void)
{
    printf("%d\n", fib(4));
    return 0;
}
```

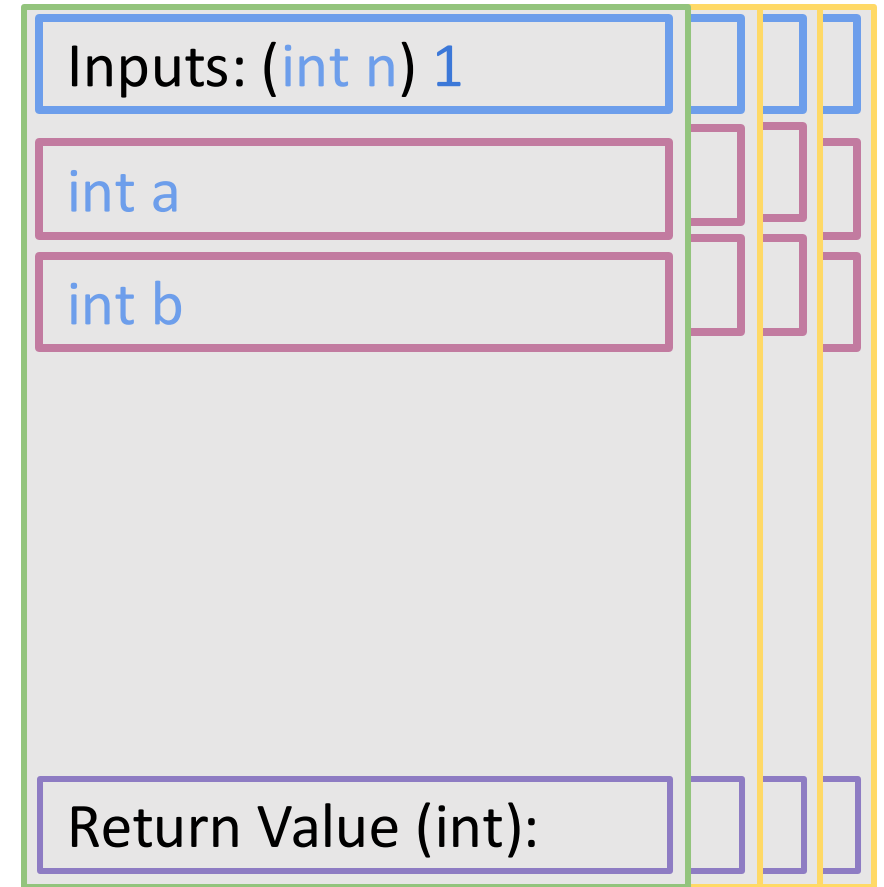


Example Function Call: Recursion

```
/* function declaration */
int fib(int n) n ← 1
{
    int a, b;

    if (n <= 1) n = 1 == 1
        return n;
    a = fib(n-1);
    b = fib(n-2);
    return a + b;
}

int main(void)
{
    printf("%d\n", fib(4));
    return 0;
}
```

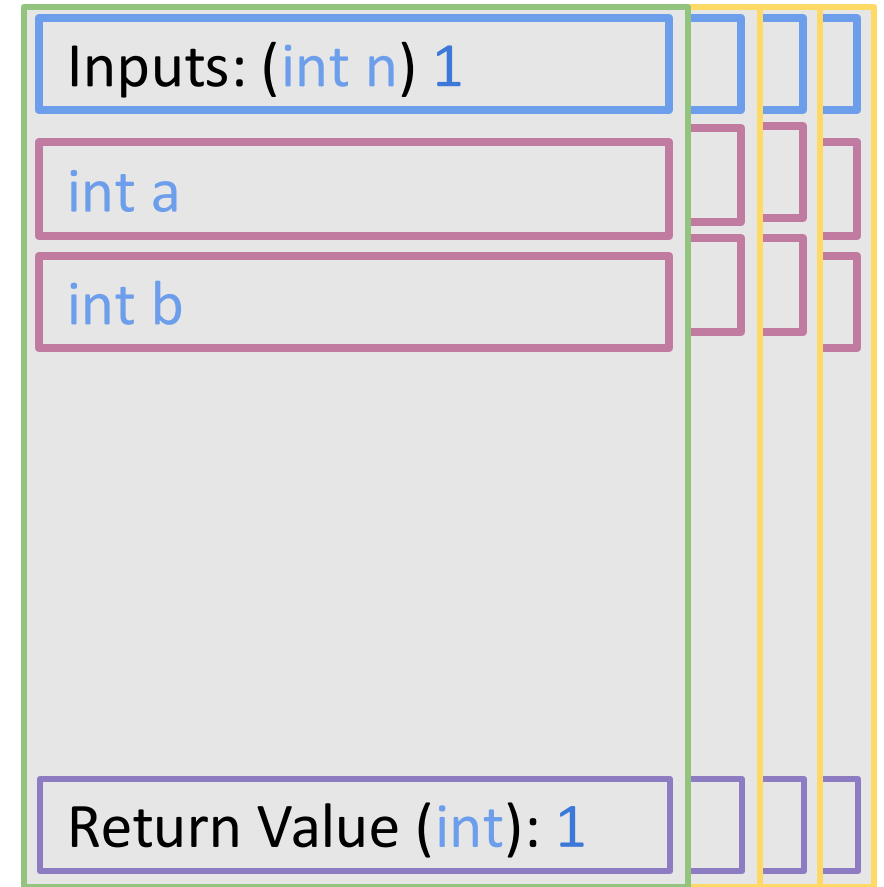


Example Function Call: Recursion

```
/* function declaration */
int fib(int n) n ← 1
{
    int a, b;

    if (n <= 1) n = 1 == 1
        return n; returns 1
    a = fib(n-1);
    b = fib(n-2);
    return a + b;
}

int main(void)
{
    printf("%d\n", fib(4));
    return 0;
}
```

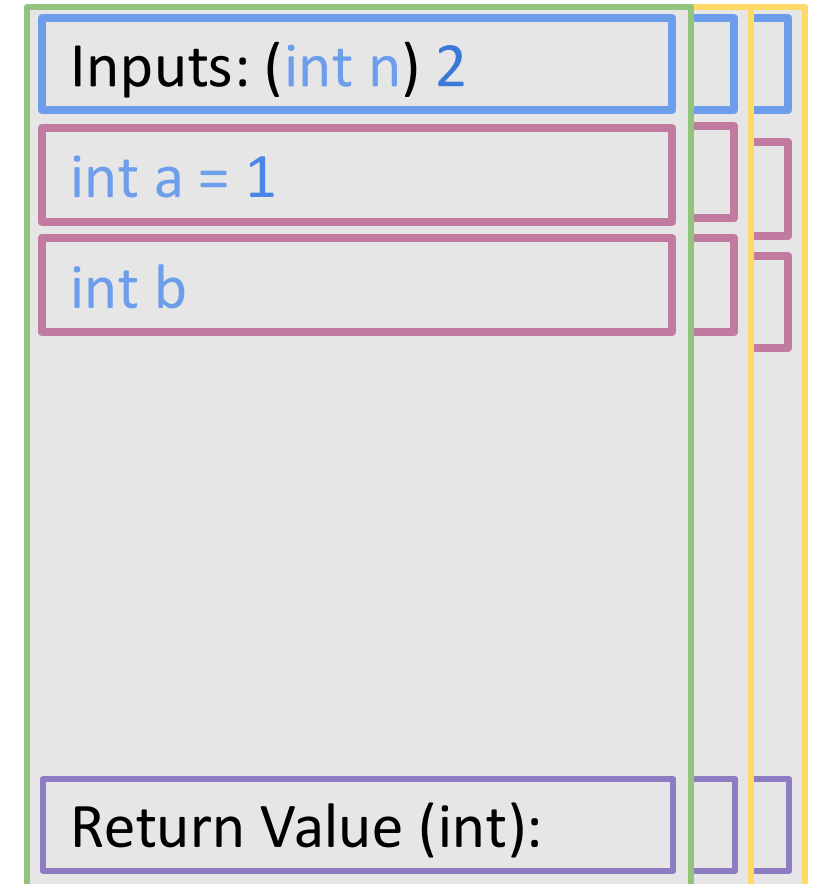


Example Function Call: Recursion

```
/* function declaration */
int fib(int n) n ← 2
{
    int a, b;

    if (n <= 1) n = 2 > 1
        return n;
    a = fib(n-1); a ← fib(2-1) = fib(1) = 1
    b = fib(n-2);
    return a + b;
}

int main(void)
{
    printf("%d\n", fib(4));
    return 0;
}
```

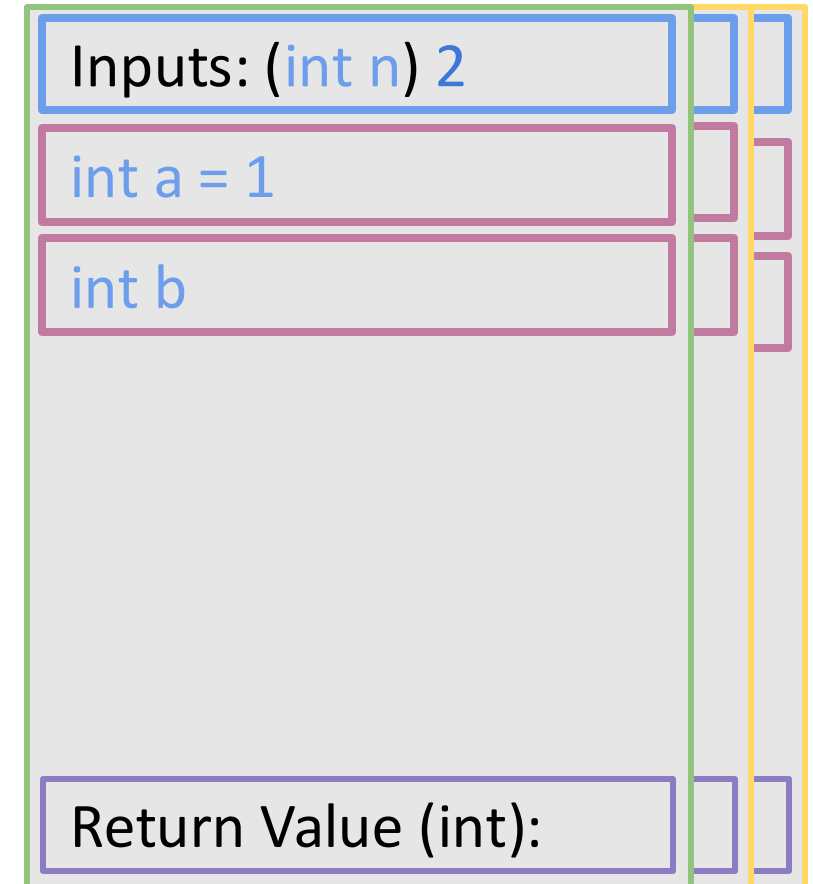


Example Function Call: Recursion

```
/* function declaration */
int fib(int n) n ← 2
{
    int a, b;

    if (n <= 1) n = 2 > 1
        return n;
    a = fib(n-1); a ← fib(2-1) = fib(1) = 1
    b = fib(n-2); b ← fib(2-2) = fib(0)
    return a + b;
}

int main(void)
{
    printf("%d\n", fib(4));
    return 0;
}
```

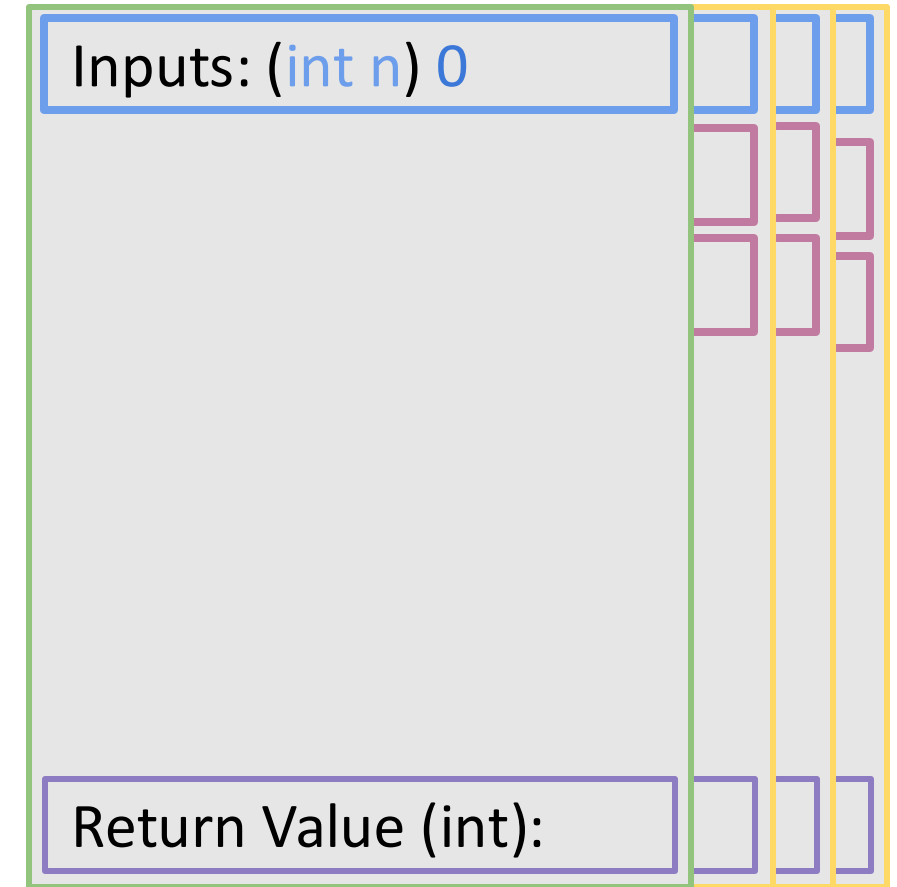


Example Function Call: Recursion

```
/* function declaration */
int fib(int n) n ← 0
{
    int a, b;

    if (n <= 1)
        return n;
    a = fib(n-1);
    b = fib(n-2);
    return a + b;
}

int main(void)
{
    printf("%d\n", fib(4));
    return 0;
}
```

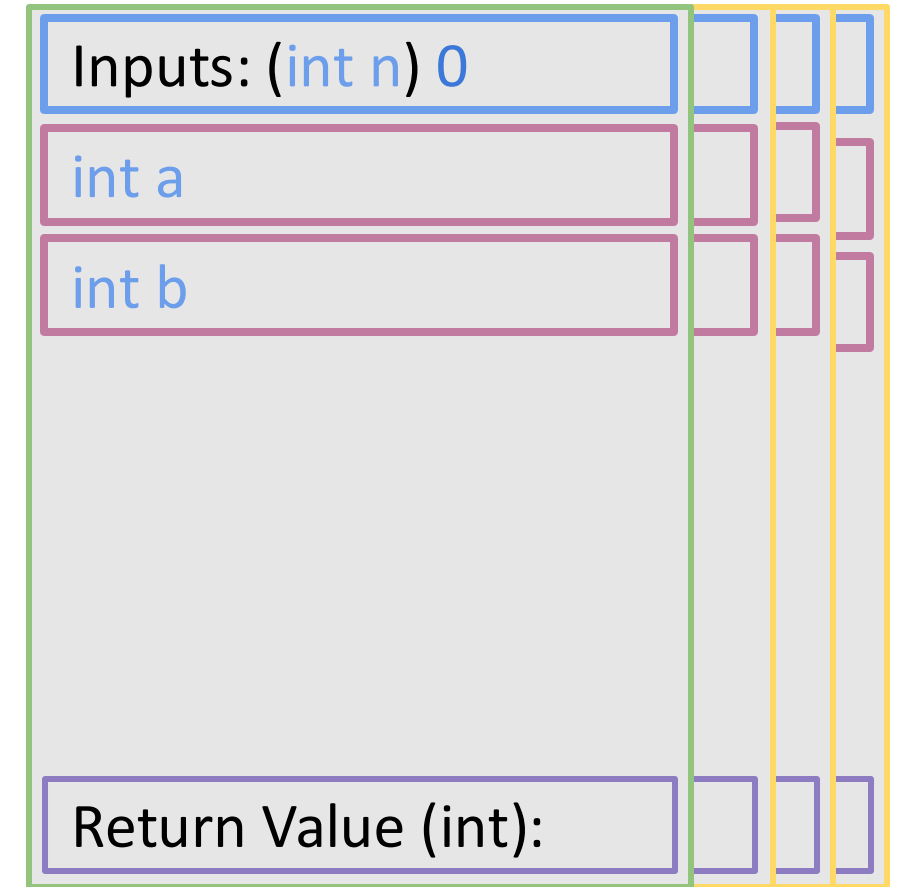


Example Function Call: Recursion

```
/* function declaration */
int fib(int n) n ← 0
{
    int a, b;

    if (n <= 1) n = 0 < 1
        return n;
    a = fib(n-1);
    b = fib(n-2);
    return a + b;
}

int main(void)
{
    printf("%d\n", fib(4));
    return 0;
}
```

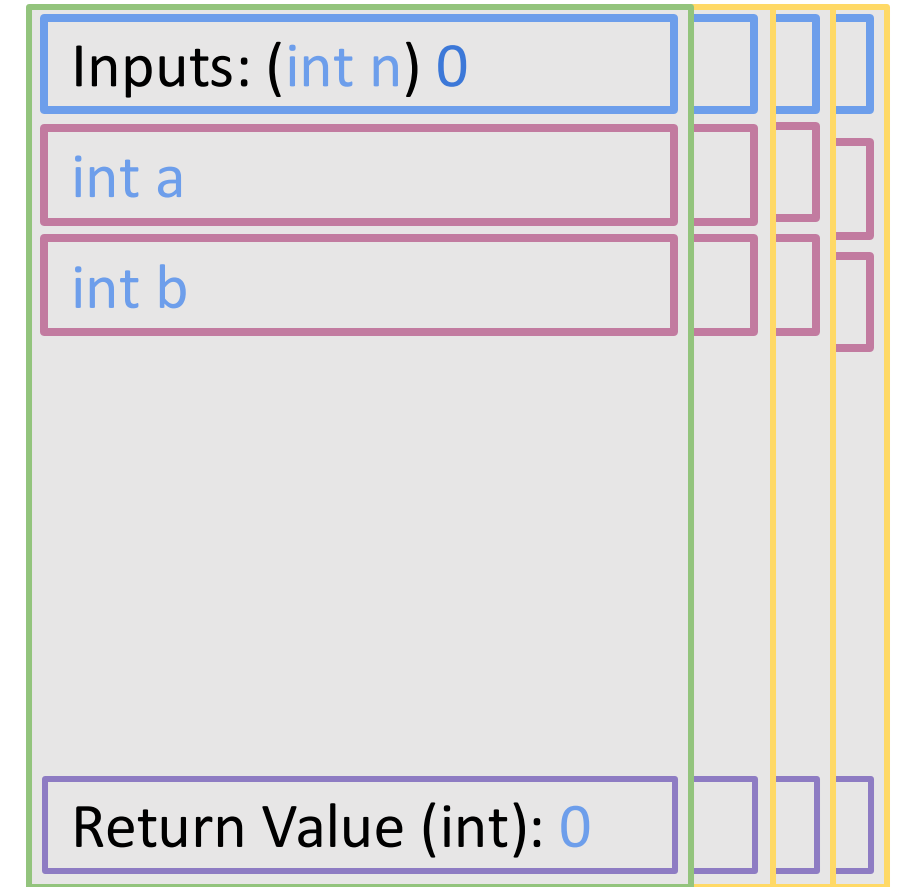


Example Function Call: Recursion

```
/* function declaration */
int fib(int n) n ← 0
{
    int a, b;

    if (n <= 1)  n = 0 < 1
        return n;  returns 0
    a = fib(n-1);
    b = fib(n-2);
    return a + b;
}

int main(void)
{
    printf("%d\n", fib(4));
    return 0;
}
```

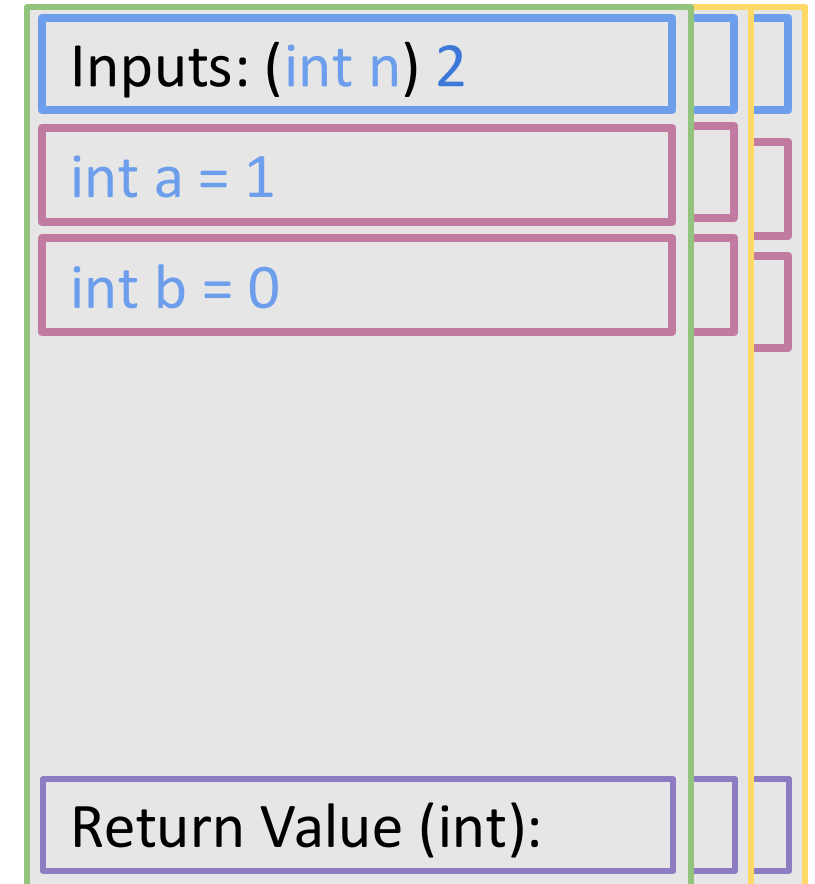


Example Function Call: Recursion

```
/* function declaration */
int fib(int n) n ← 2
{
    int a, b;

    if (n <= 1) n = 2 > 1
        return n;
    a = fib(n-1); a ← fib(2-1) = fib(1) = 1
    b = fib(n-2); b ← fib(2-2) = fib(0) = 0
    return a + b;
}

int main(void)
{
    printf("%d\n", fib(4));
    return 0;
}
```

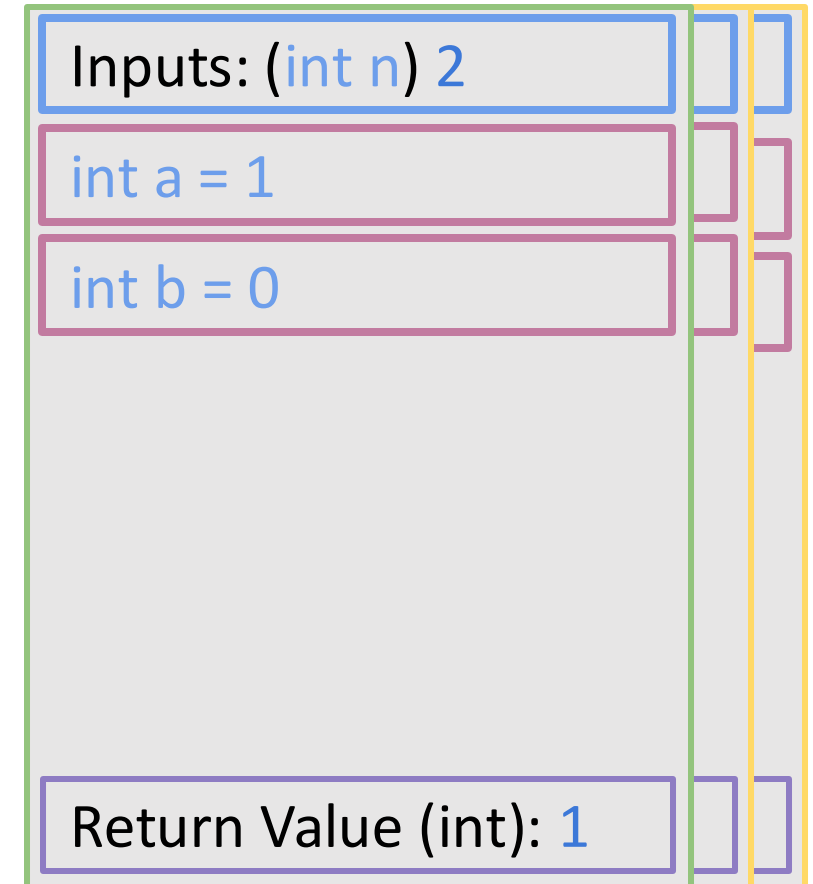


Example Function Call: Recursion

```
/* function declaration */
int fib(int n) n ← 2
{
    int a, b;

    if (n <= 1) n = 2 > 1
        return n;
    a = fib(n-1); a ← fib(2-1) = fib(1) = 1
    b = fib(n-2); b ← fib(2-2) = fib(0) = 0
    return a + b; returns 1 + 0 or 1
}

int main(void)
{
    printf("%d\n", fib(4));
    return 0;
}
```

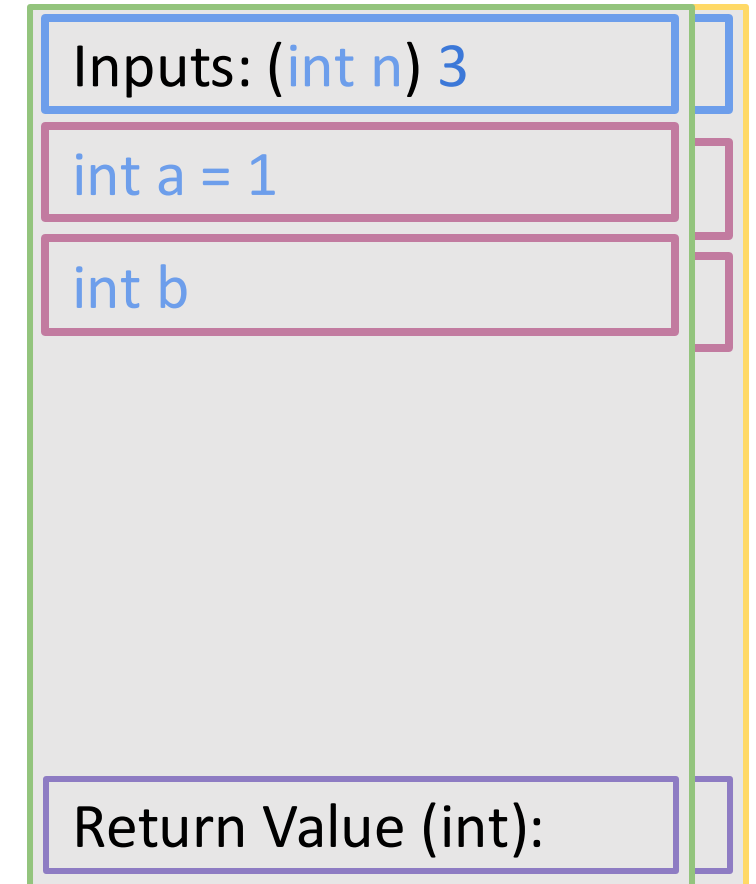


Example Function Call: Recursion

```
/* function declaration */
int fib(int n) n ← 3
{
    int a, b;

    if (n <= 1)    n == 3 > 1
        return n;
    a = fib(n-1);  a = fib(3-1) = fib(2) = 1
    b = fib(n-2);
    return a + b;
}

int main(void)
{
    printf("%d\n", fib(4));
    return 0;
}
```

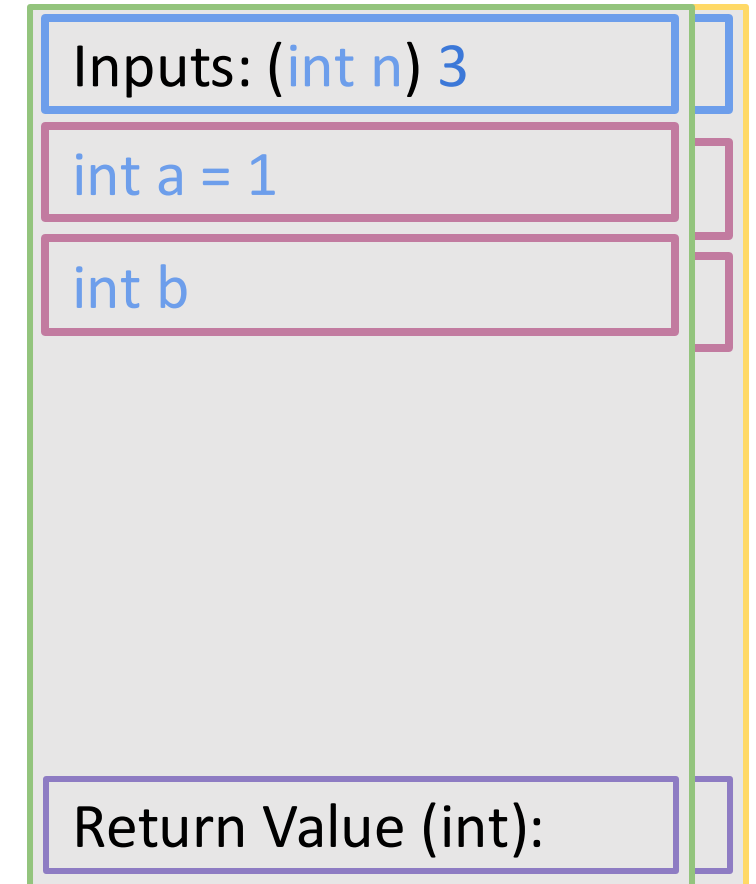


Example Function Call: Recursion

```
/* function declaration */
int fib(int n) n ← 3
{
    int a, b;

    if (n <= 1)    n == 3 > 1
        return n;
    a = fib(n-1);  a = fib(3-1) = fib(2) = 1
    b = fib(n-2);  b = fib(3-2) = fib(1)
    return a + b;
}

int main(void)
{
    printf("%d\n", fib(4));
    return 0;
}
```

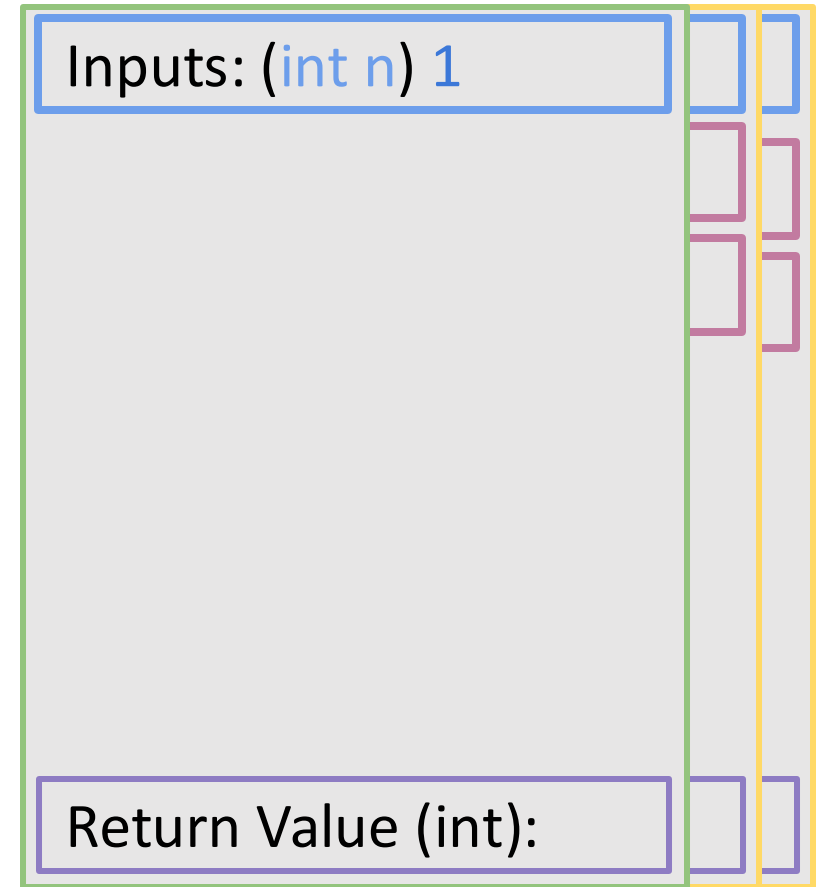


Example Function Call: Recursion

```
/* function declaration */
int fib(int n) n ← 1
{
    int a, b;

    if (n <= 1)
        return n;
    a = fib(n-1);
    b = fib(n-2);
    return a + b;
}

int main(void)
{
    printf("%d\n", fib(4));
    return 0;
}
```

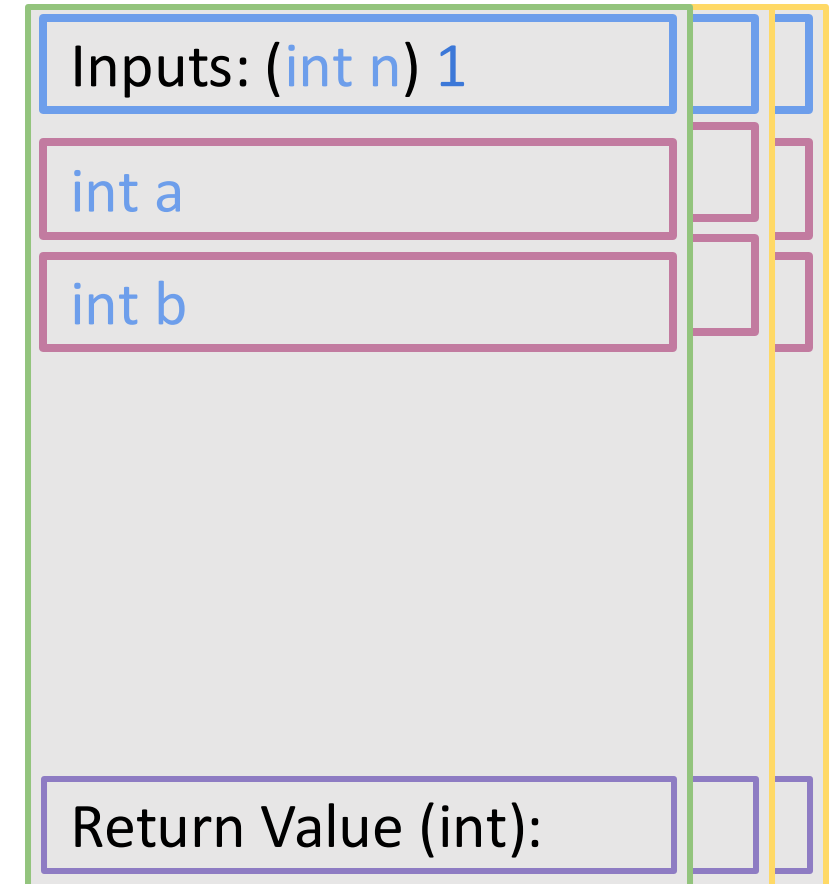


Example Function Call: Recursion

```
/* function declaration */
int fib(int n) n ← 1
{
    int a, b;

    if (n <= 1) n = 1 == 1
        return n;
    a = fib(n-1);
    b = fib(n-2);
    return a + b;
}

int main(void)
{
    printf("%d\n", fib(4));
    return 0;
}
```

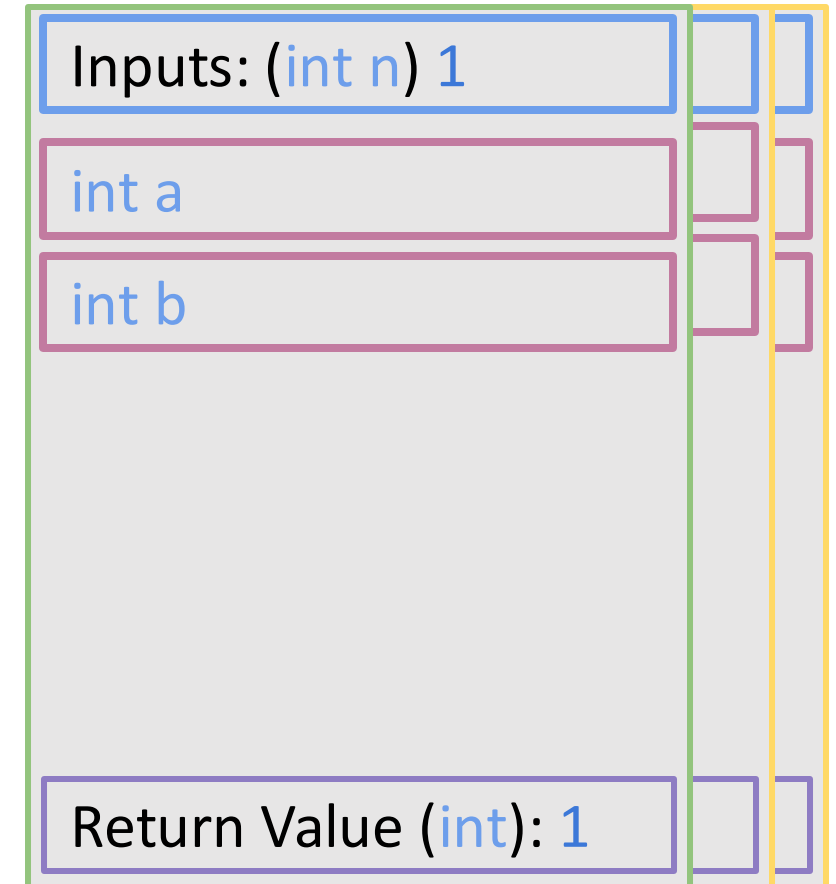


Example Function Call: Recursion

```
/* function declaration */
int fib(int n) n ← 1
{
    int a, b;

    if (n <= 1) n = 1 == 1
        return n; returns 1
    a = fib(n-1);
    b = fib(n-2);
    return a + b;
}

int main(void)
{
    printf("%d\n", fib(4));
    return 0;
}
```

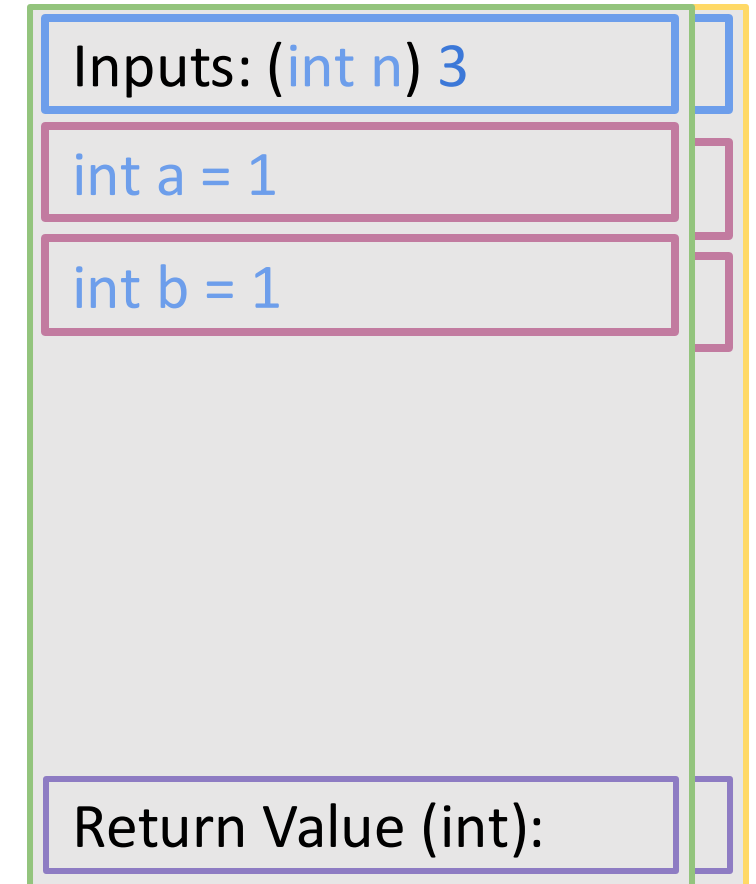


Example Function Call: Recursion

```
/* function declaration */
int fib(int n) n ← 3
{
    int a, b;

    if (n <= 1)    n == 3 > 1
        return n;
    a = fib(n-1);  a = fib(3-1) = fib(2) = 1
    b = fib(n-2);  b = fib(3-2) = fib(1) = 1
    return a + b;
}

int main(void)
{
    printf("%d\n", fib(4));
    return 0;
}
```

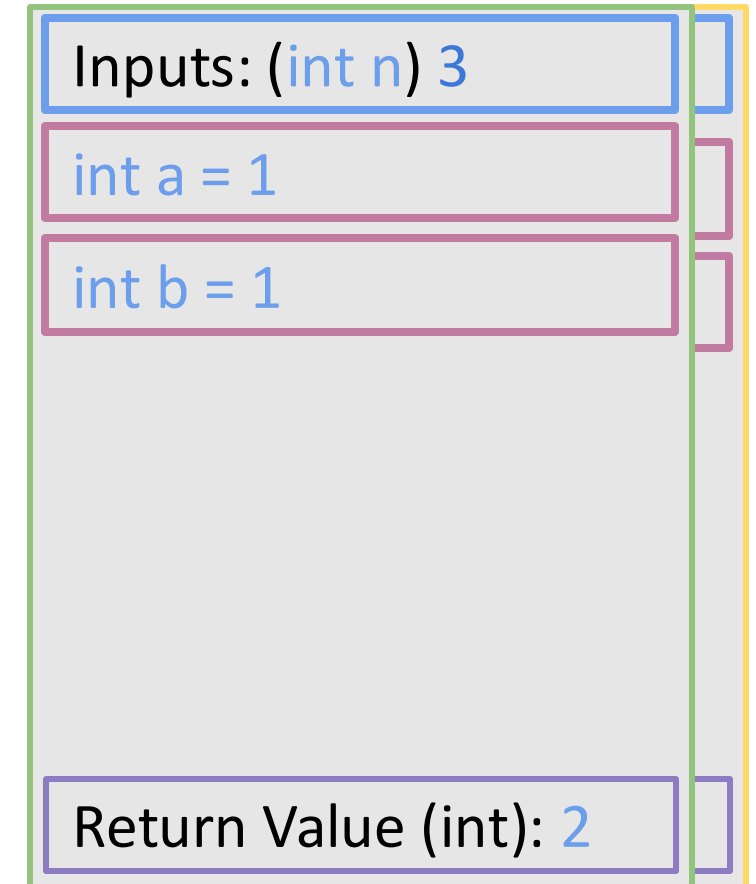


Example Function Call: Recursion

```
/* function declaration */
int fib(int n) n ← 3
{
    int a, b;

    if (n <= 1)    n == 3 > 1
        return n;
    a = fib(n-1);  a = fib(3-1) = fib(2) = 1
    b = fib(n-2);  b = fib(3-2) = fib(1) = 1
    return a + b;  return 1 + 1 or 2
}

int main(void)
{
    printf("%d\n", fib(4));
    return 0;
}
```



Example Function Call: Recursion

```
/* function declaration */
int fib(int n) n ← 4
{
    int a, b;

    if (n <= 1)    n == 4 > 1
        return n;
    a = fib(n-1);  a = fib(4-1) = fib(3) = 2
    b = fib(n-2);
    return a + b;
}

int main(void)
{
    printf("%d\n", fib(4));
    return 0;
}
```

Inputs: (int n) 4

int a = 2

int b

Return Value (int):

Example Function Call: Recursion

```
/* function declaration */
int fib(int n) n ← 4
{
    int a, b;

    if (n <= 1)    n == 4 > 1
        return n;
    a = fib(n-1);  a = fib(4-1) = fib(3) = 2
    b = fib(n-2);  b = fib(4-2) = fib(2)
    return a + b;
}

int main(void)
{
    printf("%d\n", fib(4));
    return 0;
}
```

Inputs: (int n) 4

int a = 2

int b

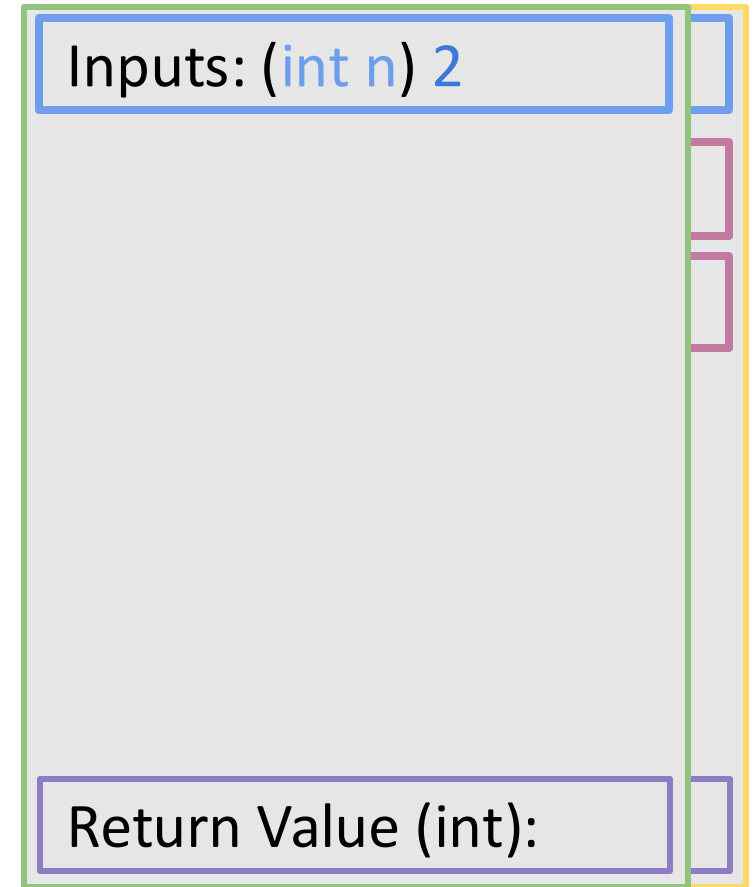
Return Value (int):

Example Function Call: Recursion

```
/* function declaration */
int fib(int n) n ← 2
{
    int a, b;

    if (n <= 1)
        return n;
    a = fib(n-1);
    b = fib(n-2);
    return a + b;
}

int main(void)
{
    printf("%d\n", fib(4));
    return 0;
}
```

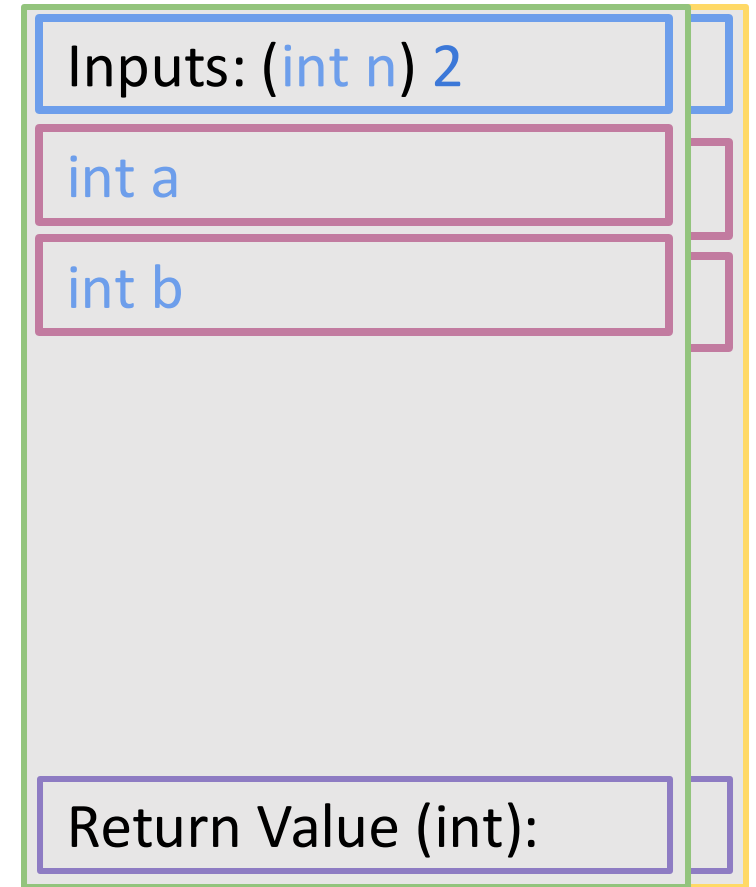


Example Function Call: Recursion

```
/* function declaration */
int fib(int n) n ← 2
{
    int a, b;

    if (n <= 1) n = 2 > 1
        return n;
    a = fib(n-1);
    b = fib(n-2);
    return a + b;
}

int main(void)
{
    printf("%d\n", fib(4));
    return 0;
}
```

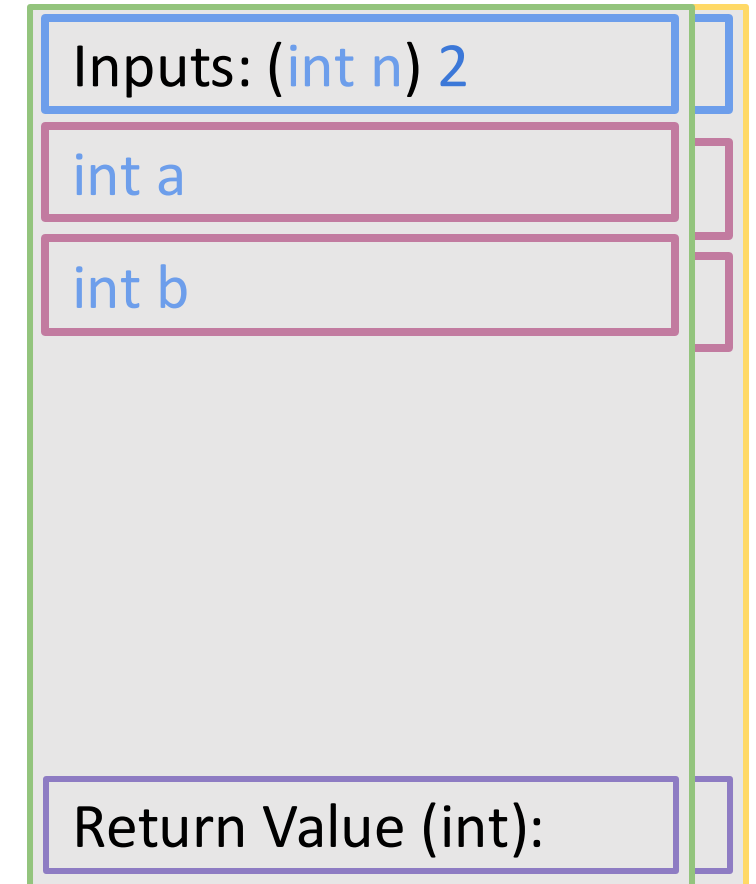


Example Function Call: Recursion

```
/* function declaration */
int fib(int n) n ← 2
{
    int a, b;

    if (n <= 1) n = 2 > 1
        return n;
    a = fib(n-1); a = fib(2-1) = fib(1)
    b = fib(n-2);
    return a + b;
}

int main(void)
{
    printf("%d\n", fib(4));
    return 0;
}
```

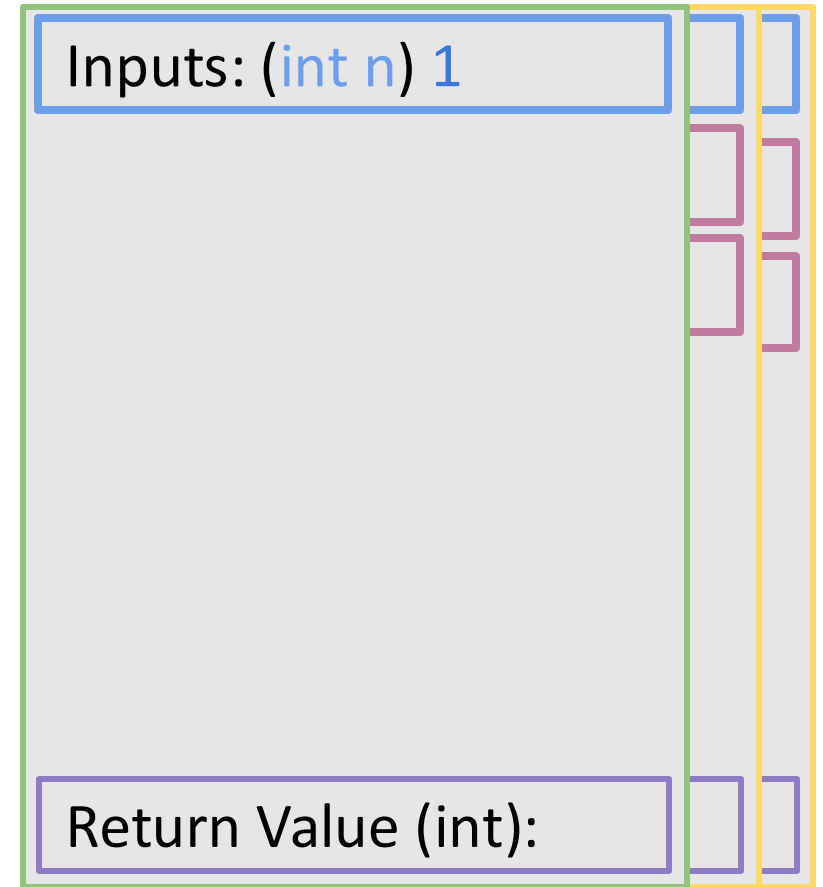


Example Function Call: Recursion

```
/* function declaration */
int fib(int n) n ← 1
{
    int a, b;

    if (n <= 1)
        return n;
    a = fib(n-1);
    b = fib(n-2);
    return a + b;
}

int main(void)
{
    printf("%d\n", fib(4));
    return 0;
}
```

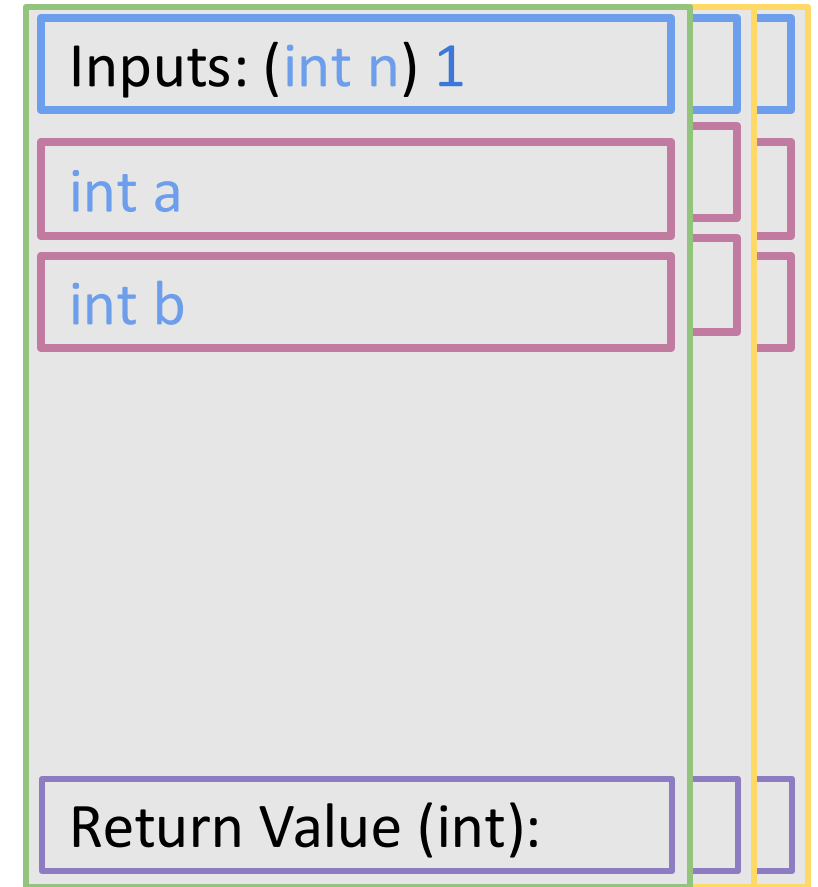


Example Function Call: Recursion

```
/* function declaration */
int fib(int n) n ← 1
{
    int a, b;

    if (n <= 1) n = 1 == 1
        return n;
    a = fib(n-1);
    b = fib(n-2);
    return a + b;
}

int main(void)
{
    printf("%d\n", fib(4));
    return 0;
}
```

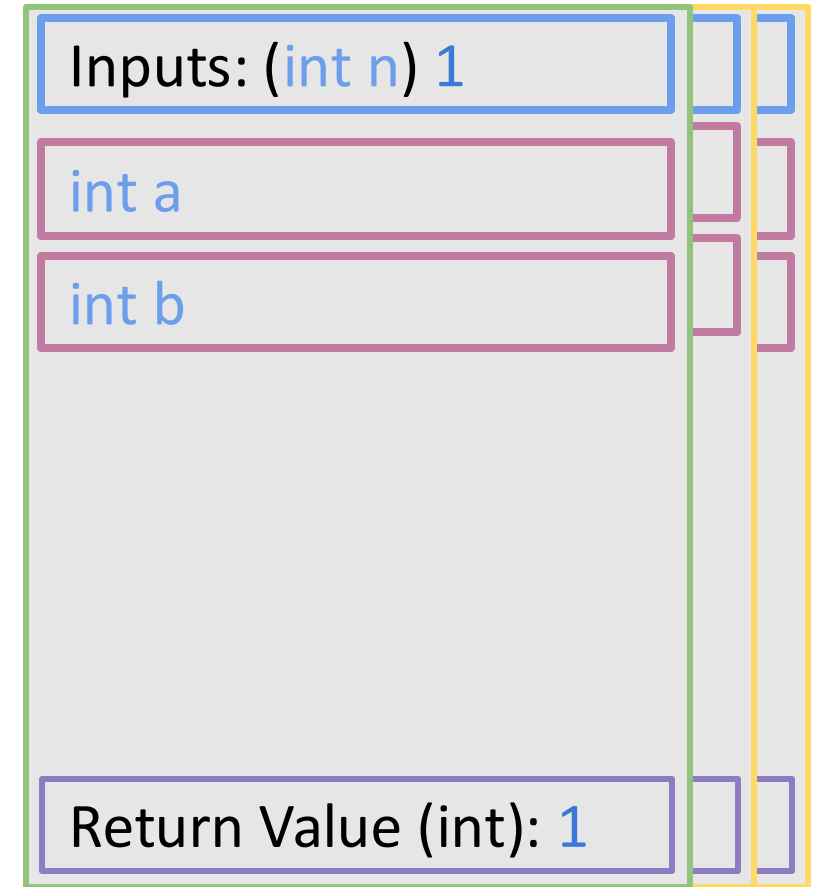


Example Function Call: Recursion

```
/* function declaration */
int fib(int n) n ← 1
{
    int a, b;

    if (n <= 1) n = 1 == 1
        return n; returns 1
    a = fib(n-1);
    b = fib(n-2);
    return a + b;
}

int main(void)
{
    printf("%d\n", fib(4));
    return 0;
}
```

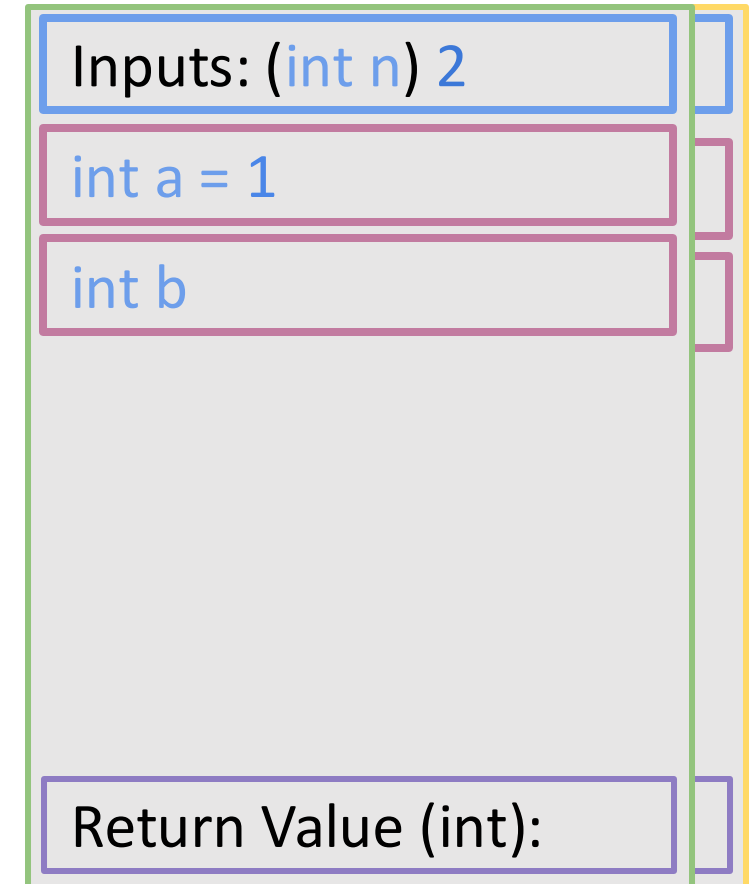


Example Function Call: Recursion

```
/* function declaration */
int fib(int n) n ← 2
{
    int a, b;

    if (n <= 1) n = 2 > 1
        return n;
    a = fib(n-1); a = fib(2-1) = fib(1) = 1
    b = fib(n-2);
    return a + b;
}

int main(void)
{
    printf("%d\n", fib(4));
    return 0;
}
```

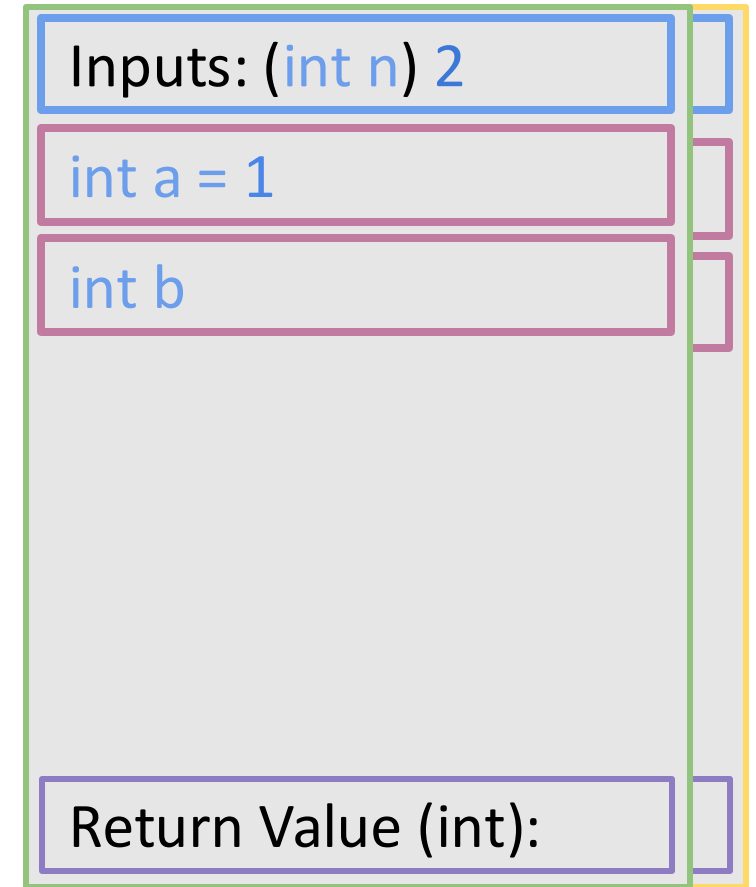


Example Function Call: Recursion

```
/* function declaration */
int fib(int n) n ← 2
{
    int a, b;

    if (n <= 1) n = 2 > 1
        return n;
    a = fib(n-1); a = fib(2-1) = fib(1) = 1
    b = fib(n-2); b = fib(2-2) = fib(0)
    return a + b;
}

int main(void)
{
    printf("%d\n", fib(4));
    return 0;
}
```

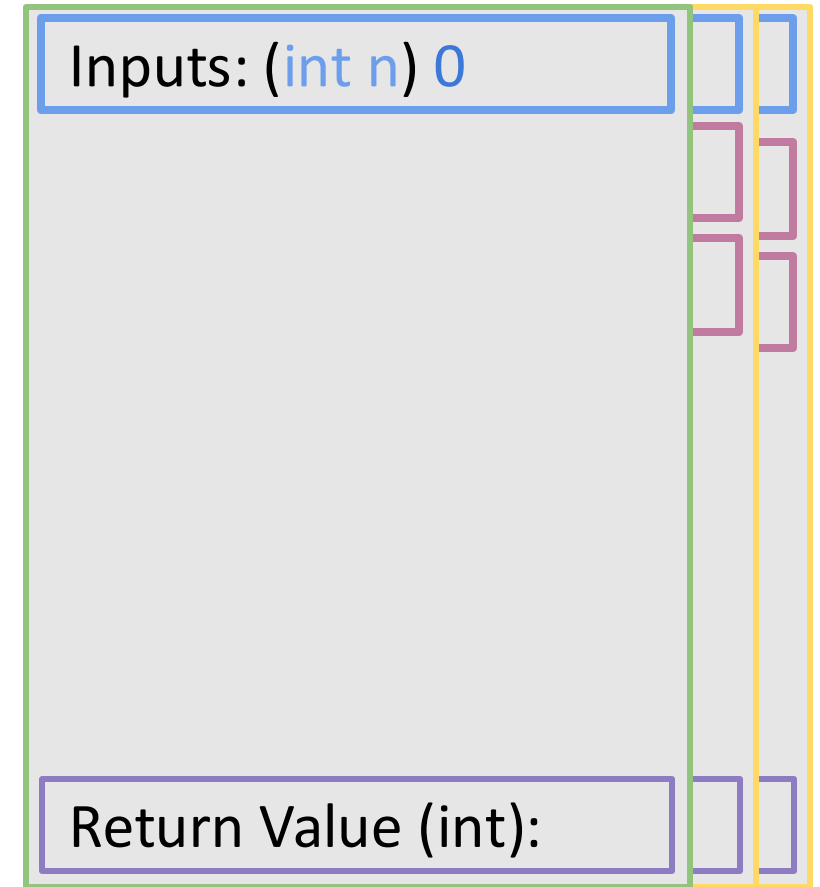


Example Function Call: Recursion

```
/* function declaration */
int fib(int n) n ← 0
{
    int a, b;

    if (n <= 1)
        return n;
    a = fib(n-1);
    b = fib(n-2);
    return a + b;
}

int main(void)
{
    printf("%d\n", fib(4));
    return 0;
}
```

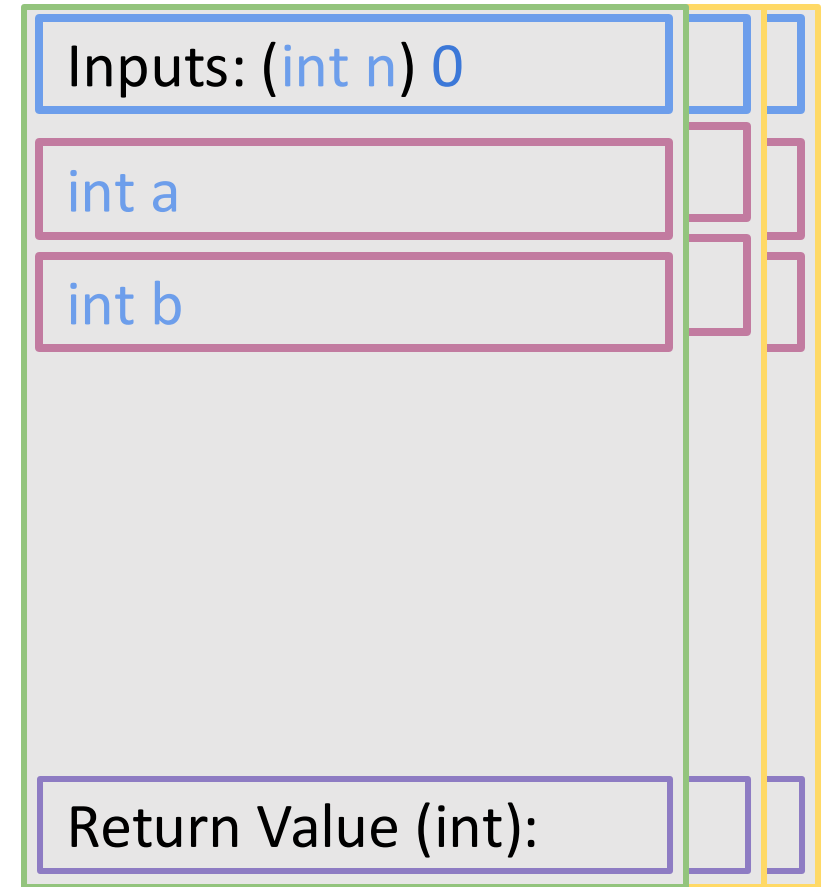


Example Function Call: Recursion

```
/* function declaration */
int fib(int n) n ← 0
{
    int a, b;

    if (n <= 1) n = 0 < 1
        return n;
    a = fib(n-1);
    b = fib(n-2);
    return a + b;
}

int main(void)
{
    printf("%d\n", fib(4));
    return 0;
}
```

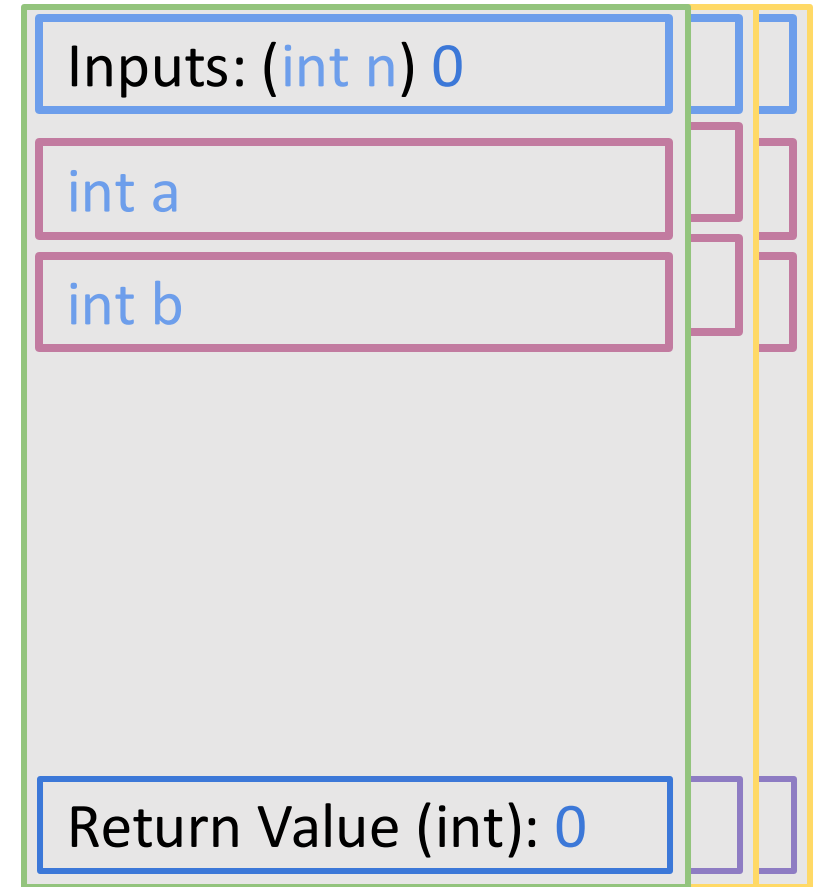


Example Function Call: Recursion

```
/* function declaration */
int fib(int n) n ← 0
{
    int a, b;

    if (n <= 1)    n = 0 < 1
        return n;    returns 0
    a = fib(n-1);
    b = fib(n-2);
    return a + b;
}

int main(void)
{
    printf("%d\n", fib(4));
    return 0;
}
```

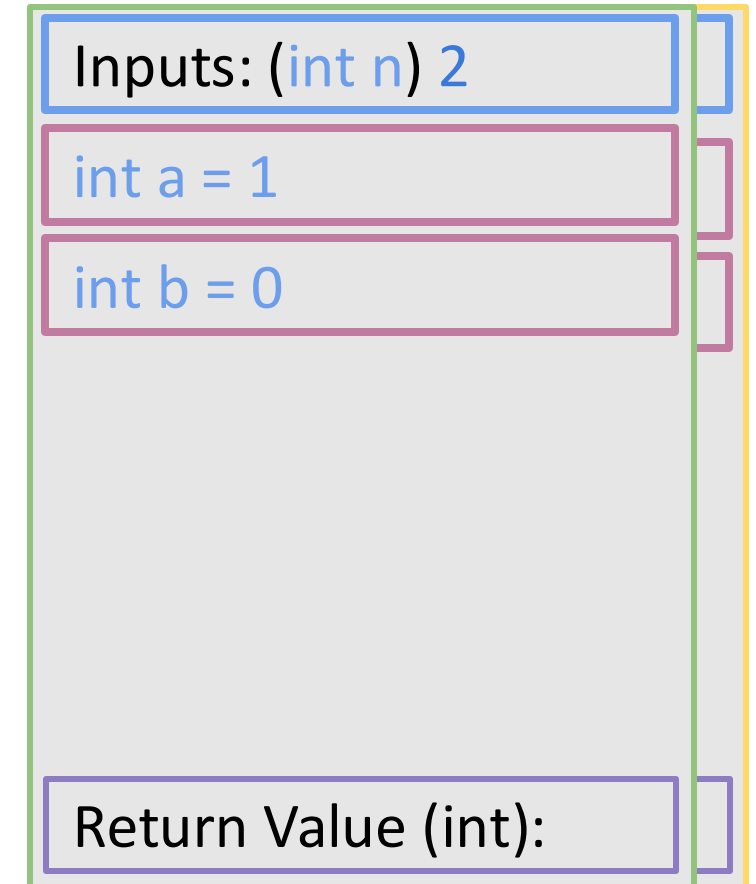


Example Function Call: Recursion

```
/* function declaration */
int fib(int n) n ← 2
{
    int a, b;

    if (n <= 1) n = 2 > 1
        return n;
    a = fib(n-1); a = fib(2-1) = fib(1) = 1
    b = fib(n-2); b = fib(2-2) = fib(0) = 0
    return a + b;
}

int main(void)
{
    printf("%d\n", fib(4));
    return 0;
}
```

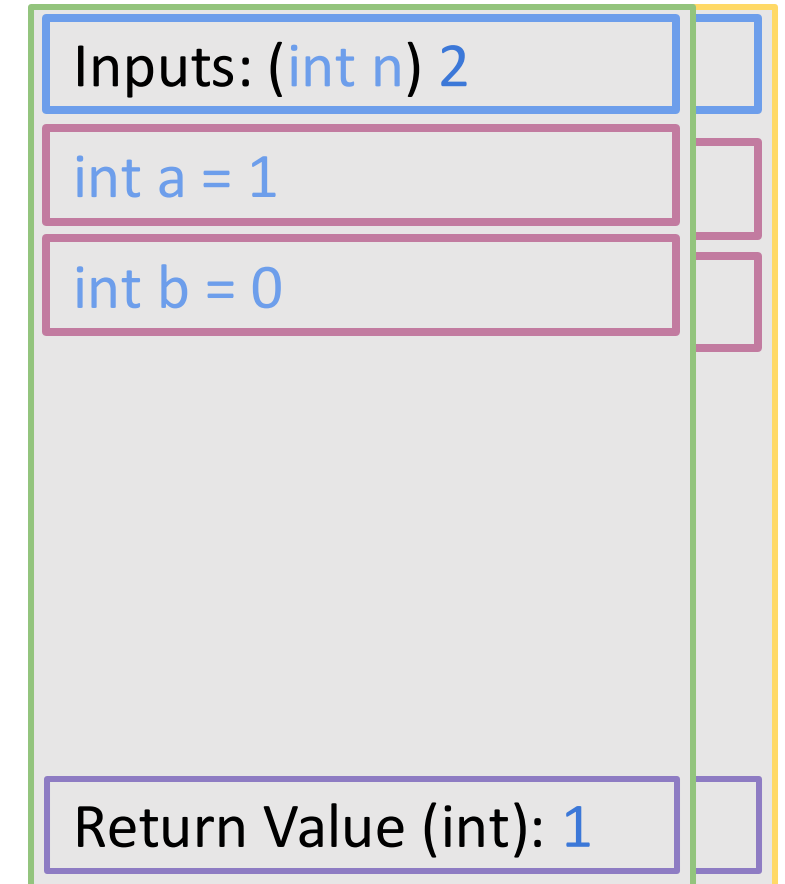


Example Function Call: Recursion

```
/* function declaration */
int fib(int n) n ← 2
{
    int a, b;

    if (n <= 1)    n = 2 > 1
        return n;
    a = fib(n-1);  a = fib(2-1) = fib(1) = 1
    b = fib(n-2);  b = fib(2-2) = fib(0) = 0
    return a + b;  returns 1 + 0 = 1
}

int main(void)
{
    printf("%d\n", fib(4));
    return 0;
}
```



Example Function Call: Recursion

```
/* function declaration */
int fib(int n) n ← 4
{
    int a, b;

    if (n <= 1)    n == 4 > 1
        return n;
    a = fib(n-1);  a = fib(4-1) = fib(3) = 2
    b = fib(n-2);  b = fib(4-2) = fib(2) = 1
    return a + b;
}

int main(void)
{
    printf("%d\n", fib(4));
    return 0;
}
```

Inputs: (int n) 4

int a = 2

int b = 1

Return Value (int):

Example Function Call: Recursion

```
/* function declaration */
int fib(int n) n ← 4
{
    int a, b;

    if (n <= 1)    n == 4 > 1
        return n;
    a = fib(n-1);  a = fib(4-1) = fib(3) = 2
    b = fib(n-2);  b = fib(4-2) = fib(2) = 1
    return a + b;  returns 2 + 1 = 3
}

int main(void)
{
    printf("%d\n", fib(4));
    return 0;
}
```

Inputs: (int n) 4

int a = 2

int b = 1

Return Value (int): 3