

Recursive Function Calls

July 20, 2009

The Fibonacci sequence

$$a_n = a_{n-2} + a_{n-1} \quad a_1 = 1, \quad a_2 = 1$$

is a an easy to program recursive function:

1. If you are passed $n = 1$ or $n = 2$, then return 1.
2. Otherwise, “call yourself” with argument values $n - 2$ and $n - 1$.
3. Sum these results and return the value.

These diagrams show how the stack grows downward with each successive function call. Each `fib()` has its own portion of the stack to store its `n`, `nMinus2`, and `nMinus1` variables.

```
1 /** The fibonacci sequence stack. */
2 #include <iostream>
3 using namespace std;
4 int fib( int n )
5 {
6     if( n <= 2 ) {
7         return 1;
8     }
9     int nMinus2( fib(n-2) );
10    int nMinus1( fib(n-1) );
11    int r( nMinus2 + nMinus1 );
12    return r;
13 }
14 int main()
15 {
16     int f;
17     f = fib(4);
18     cout << "fibonacci (4)="
19          << f << endl;
20     return 0;
21 }
```

Although the variables are named the same, each `fib()` function “sees” only those variables in its own stack frame. For the same reason, different functions can share like-named variables without conflict.

No wonder they are called **automatic** variables. Memory to hold their values is *automatically* found on the program stack.

```
1 /** The fibonacci sequence stack. */
2 #include <iostream>
3 using namespace std;
4 int fib( int n )
5 {
6     if( n <= 2 ) {
7         return 1;
8     }
9     int nMinus2( fib(n-2) );
10    int nMinus1( fib(n-1) );
11    int r( nMinus2 + nMinus1 );
12    return r;
13 }
14 int main()
15 {
16     int f;
17     f = fib(4);
18     cout << "fibonacci(4)="
19          << f << endl;
20     return 0;
21 }
```

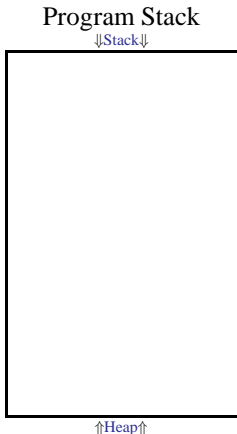
In order to make these diagrams more readable, the function `fib` is referred to along with the value of `n` it is called with. The main function calls `fib(4)`, and it then calls `fib(2)` and `fib(3)`,...

The parameter value has been added to `fib()` purely for readability. There are *not* different `fib` functions for each value of `n`.

```
1 /** The fibonacci sequence stack. */
2 #include <iostream>
3 using namespace std;
4 int fib( int n )
5 {
6     if( n <= 2 ) {
7         return 1;
8     }
9     int nMinus2( fib(n-2) );
10    int nMinus1( fib(n-1) );
11    int r( nMinus2 + nMinus1 );
12    return r;
13 }
14 int main()
15 {
16     int f;
17     f = fib(4);
18     cout << "fibonacci(4)="
19          << f << endl;
20     return 0;
21 }
```

The Operating System calls main ()

<OS>

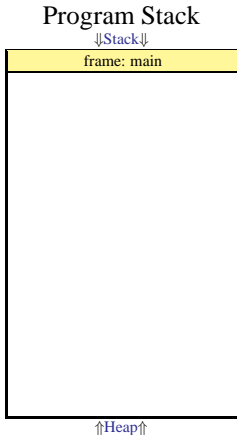


```
1 /** The fibonacci sequence stack. */
2 #include <iostream>
3 using namespace std;
4 int fib( int n )
5 {
6     if( n <= 2 ) {
7         return 1;
8     }
9     int nMinus2( fib(n-2) );
10    int nMinus1( fib(n-1) );
11    int r( nMinus2 + nMinus1 );
12    return r;
13 }
14 int main()
15 {
16     int f;
17     f = fib(4);
18     cout << "fibonacci (4)="
19         << f << endl;
20     return 0;
21 }
```

The Operating System calls `main()`

create call frame for `main`

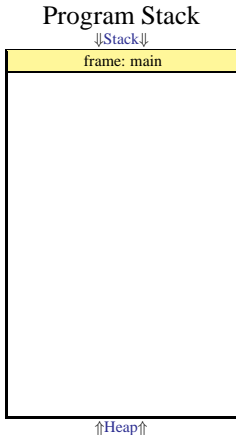
<OS>



```
1 /** The fibonacci sequence stack. */
2 #include <iostream>
3 using namespace std;
4 int fib( int n )
5 {
6     if( n <= 2 ) {
7         return 1;
8     }
9     int nMinus2( fib(n-2) );
10    int nMinus1( fib(n-1) );
11    int r( nMinus2 + nMinus1 );
12    return r;
13 }
14 int main()
15 {
16     int f;
17     f = fib(4);
18     cout << "fibonacci (4)="
19          << f << endl;
20     return 0;
21 }
```

Line 14: Function main entry

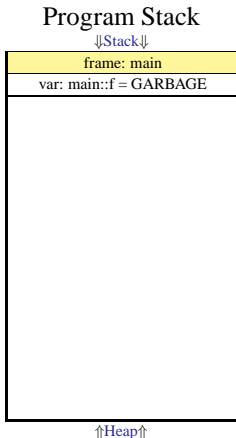
<OS> \implies main



```
1 /** The fibonacci sequence stack. */
2 #include <iostream>
3 using namespace std;
4 int fib( int n )
5 {
6     if( n <= 2 ) {
7         return 1;
8     }
9     int nMinus2( fib(n-2) );
10    int nMinus1( fib(n-1) );
11    int r( nMinus2 + nMinus1 );
12    return r;
13 }
14 int main() // <==
15 {
16     int f;
17     f = fib(4);
18     cout << "fibonacci (4)="
19         << f << endl;
20     return 0;
21 }
```


Line 17: Call fib(4)

<OS> \implies main

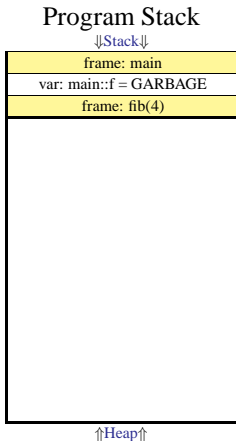


```
1  /** The fibonacci sequence stack. */
2  #include <iostream>
3  using namespace std;
4  int fib( int n )
5  {
6      if( n <= 2 ) {
7          return 1;
8      }
9      int nMinus2( fib(n-2) );
10     int nMinus1( fib(n-1) );
11     int r( nMinus2 + nMinus1 );
12     return r;
13 }
14 int main()
15 {
16     int f;
17     f = fib(4);
18     cout << "fibonacci(4)="
19          << f << endl;
20     return 0;
21 }
```

Line 17: Call fib(4)

create call frame for fib(4)

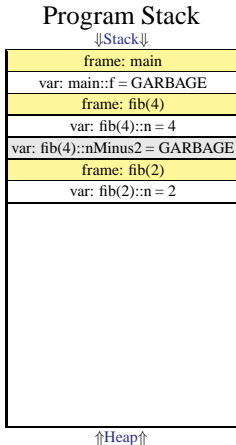
<OS> \implies main



```
1  /** The fibonacci sequence stack. */
2  #include <iostream>
3  using namespace std;
4  int fib( int n )
5  {
6      if( n <= 2 ) {
7          return 1;
8      }
9      int nMinus2( fib(n-2) );
10     int nMinus1( fib(n-1) );
11     int r( nMinus2 + nMinus1 );
12     return r;
13 }
14 int main()
15 {
16     int f;
17     f = fib(4);
18     cout << "fibonacci(4)="
19          << f << endl;
20     return 0;
21 }
```


Line 4: Function fib(2) entry

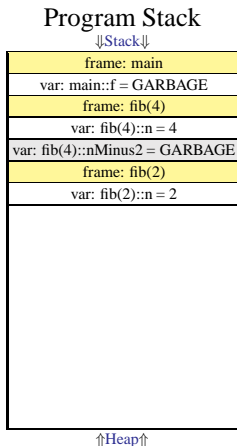
<OS> \implies main \implies fib(4) \implies fib(2)



```
1 /** The fibonacci sequence stack. */
2 #include <iostream>
3 using namespace std;
4 int fib( int n )           // <==
5 {
6     if( n <= 2 ) {
7         return 1;
8     }
9     int nMinus2( fib(n-2) );
10    int nMinus1( fib(n-1) );
11    int r( nMinus2 + nMinus1 );
12    return r;
13 }
14 int main()
15 {
16     int f;
17     f = fib(4);
18     cout << "fibonacci (4)="
19          << f << endl;
20     return 0;
21 }
```


Line 7: Yes, n is 2, return!

<OS> \implies main \implies fib(4) \implies fib(2)

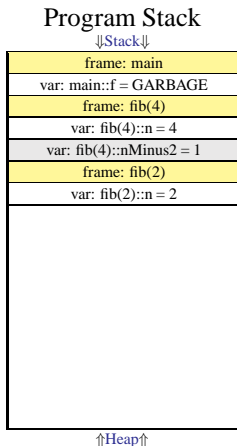


```
1  /** The fibonacci sequence stack. */
2  #include <iostream>
3  using namespace std;
4  int fib( int n )
5  {
6      if( n <= 2 ) {
7          return 1;           // <==
8      }
9      int nMinus2( fib(n-2) );
10     int nMinus1( fib(n-1) );
11     int r( nMinus2 + nMinus1 );
12     return r;
13 }
14 int main()
15 {
16     int f;
17     f = fib(4);
18     cout << "fibonacci (4)="
19          << f << endl;
20     return 0;
21 }
```

Line 7: Yes, n is 2, return!

copy return value

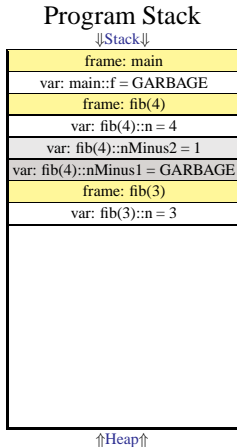
<OS> \implies main \implies fib(4) \implies fib(2)



```
1 /** The fibonacci sequence stack. */
2 #include <iostream>
3 using namespace std;
4 int fib( int n )
5 {
6     if( n <= 2 ) {
7         return 1;           // <==
8     }
9     int nMinus2( fib(n-2) );
10    int nMinus1( fib(n-1) );
11    int r( nMinus2 + nMinus1 );
12    return r;
13 }
14 int main()
15 {
16     int f;
17     f = fib(4);
18     cout << "fibonacci (4)="
19          << f << endl;
20     return 0;
21 }
```


Line 4: Function fib(3) entry

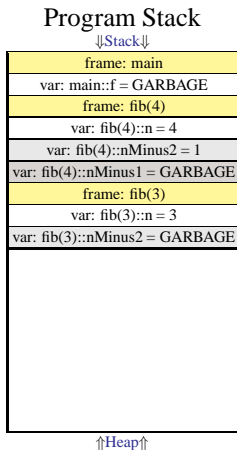
<OS> \implies main \implies fib(4) \implies fib(3)



```
1 /** The fibonacci sequence stack. */
2 #include <iostream>
3 using namespace std;
4 int fib( int n )           // <==
5 {
6     if( n <= 2 ) {
7         return 1;
8     }
9     int nMinus2( fib(n-2) );
10    int nMinus1( fib(n-1) );
11    int r( nMinus2 + nMinus1 );
12    return r;
13 }
14 int main()
15 {
16     int f;
17     f = fib(4);
18     cout << "fibonacci (4)="
19          << f << endl;
20     return 0;
21 }
```


Line 9: No, $n > 2$, Declare `nMinus2`...

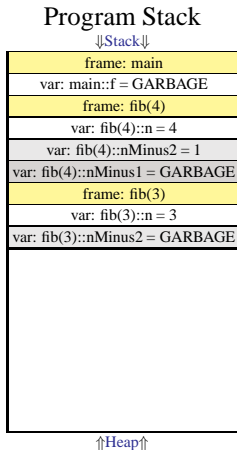
`<OS>` \implies `main` \implies `fib(4)` \implies `fib(3)`



```
1 /** The fibonacci sequence stack. */
2 #include <iostream>
3 using namespace std;
4 int fib( int n )
5 {
6     if( n <= 2 ) {
7         return 1;
8     }
9     int nMinus2( fib(n-2) );    // <==
10    int nMinus1( fib(n-1) );
11    int r( nMinus2 + nMinus1 );
12    return r;
13 }
14 int main()
15 {
16     int f;
17     f = fib(4);
18     cout << "fibonacci (4)="
19          << f << endl;
20     return 0;
21 }
```

Line 9: Call fib(1)

<OS> \implies main \implies fib(4) \implies fib(3)

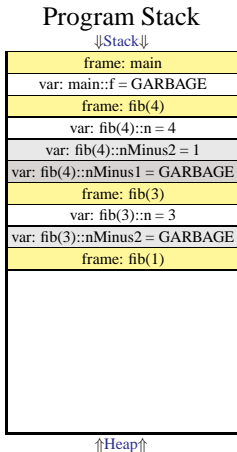


```
1  /** The fibonacci sequence stack. */
2  #include <iostream>
3  using namespace std;
4  int fib( int n )
5  {
6      if( n <= 2 ) {
7          return 1;
8      }
9      int nMinus2( fib(n-2) );    // <==
10     int nMinus1( fib(n-1) );
11     int r( nMinus2 + nMinus1 );
12     return r;
13 }
14 int main()
15 {
16     int f;
17     f = fib(4);
18     cout << "fibonacci(4)="
19          << f << endl;
20     return 0;
21 }
```

Line 9: Call fib(1)

create call frame for fib(1)

<OS> \implies main \implies fib(4) \implies fib(3)

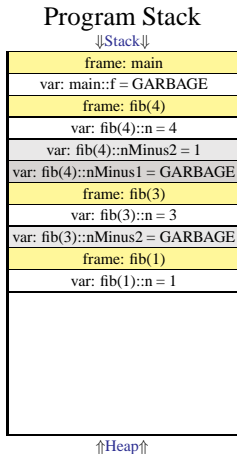


```
1  /** The fibonacci sequence stack. */
2  #include <iostream>
3  using namespace std;
4  int fib( int n )
5  {
6      if( n <= 2 ) {
7          return 1;
8      }
9      int nMinus2( fib(n-2) );    // <==
10     int nMinus1( fib(n-1) );
11     int r( nMinus2 + nMinus1 );
12     return r;
13 }
14 int main()
15 {
16     int f;
17     f = fib(4);
18     cout << "fibonacci(4)="
19          << f << endl;
20     return 0;
21 }
```

Line 9: Call fib(1)

create formal param n of fib(1)

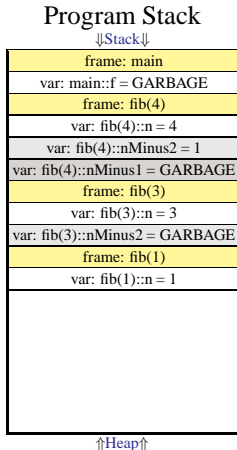
<OS> \implies main \implies fib(4) \implies fib(3)



```
1  /** The fibonacci sequence stack. */
2  #include <iostream>
3  using namespace std;
4  int fib( int n )
5  {
6      if( n <= 2 ) {
7          return 1;
8      }
9      int nMinus2( fib(n-2) );    // <==
10     int nMinus1( fib(n-1) );
11     int r( nMinus2 + nMinus1 );
12     return r;
13 }
14 int main()
15 {
16     int f;
17     f = fib(4);
18     cout << "fibonacci(4)="
19          << f << endl;
20     return 0;
21 }
```

Line 4: Function fib(1) entry

<OS> \implies main \implies fib(4) \implies fib(3) \implies fib(1)



```
1 /** The fibonacci sequence stack. */
2 #include <iostream>
3 using namespace std;
4 int fib( int n )           // <==
5 {
6     if( n <= 2 ) {
7         return 1;
8     }
9     int nMinus2( fib(n-2) );
10    int nMinus1( fib(n-1) );
11    int r( nMinus2 + nMinus1 );
12    return r;
13 }
14 int main()
15 {
16     int f;
17     f = fib(4);
18     cout << "fibonacci (4)="
19          << f << endl;
20     return 0;
21 }
```

Line 6: Is n==1 or n==2?

<OS> \implies main \implies fib(4) \implies fib(3) \implies fib(1)

Program Stack

↓Stack↓

frame: main
var: main::f = GARBAGE
frame: fib(4)
var: fib(4)::n = 4
var: fib(4)::nMinus2 = 1
var: fib(4)::nMinus1 = GARBAGE
frame: fib(3)
var: fib(3)::n = 3
var: fib(3)::nMinus2 = GARBAGE
frame: fib(1)
var: fib(1)::n = 1

↑Heap↑

```
1  /** The fibonacci sequence stack. */
2  #include <iostream>
3  using namespace std;
4  int fib( int n )
5  {
6      if( n <= 2 ) {                // <==
7          return 1;
8      }
9      int nMinus2( fib(n-2) );
10     int nMinus1( fib(n-1) );
11     int r( nMinus2 + nMinus1 );
12     return r;
13 }
14 int main()
15 {
16     int f;
17     f = fib(4);
18     cout << "fibonacci(4)="
19          << f << endl;
20     return 0;
21 }
```

Line 7: Yes, n is 1, return!

<OS> \implies main \implies fib(4) \implies fib(3) \implies fib(1)

Program Stack

↓Stack↓

frame: main
var: main::f = GARBAGE
frame: fib(4)
var: fib(4)::n = 4
var: fib(4)::nMinus2 = 1
var: fib(4)::nMinus1 = GARBAGE
frame: fib(3)
var: fib(3)::n = 3
var: fib(3)::nMinus2 = GARBAGE
frame: fib(1)
var: fib(1)::n = 1

↑Heap↑

```
1 /** The fibonacci sequence stack. */
2 #include <iostream>
3 using namespace std;
4 int fib( int n )
5 {
6     if( n <= 2 ) {
7         return 1;           // <==
8     }
9     int nMinus2( fib(n-2) );
10    int nMinus1( fib(n-1) );
11    int r( nMinus2 + nMinus1 );
12    return r;
13 }
14 int main()
15 {
16     int f;
17     f = fib(4);
18     cout << "fibonacci (4)="
19          << f << endl;
20     return 0;
21 }
```

Line 7: Yes, n is 1, return!

copy return value

<OS> \implies main \implies fib(4) \implies fib(3) \implies fib(1)

Program Stack

↓Stack↓

frame: main
var: main::f = GARBAGE
frame: fib(4)
var: fib(4)::n = 4
var: fib(4)::nMinus2 = 1
var: fib(4)::nMinus1 = GARBAGE
frame: fib(3)
var: fib(3)::n = 3
var: fib(3)::nMinus2 = 1
frame: fib(1)
var: fib(1)::n = 1

↑Heap↑

```
1 /** The fibonacci sequence stack. */
2 #include <iostream>
3 using namespace std;
4 int fib( int n )
5 {
6     if ( n <= 2 ) {
7         return 1;           // <==
8     }
9     int nMinus2( fib(n-2) );
10    int nMinus1( fib(n-1) );
11    int r( nMinus2 + nMinus1 );
12    return r;
13 }
14 int main()
15 {
16     int f;
17     f = fib(4);
18     cout << "fibonacci (4)="
19          << f << endl;
20     return 0;
21 }
```

Line 7: Yes, n is 1, return!

discard fib(1) parameter n

<OS> \implies main \implies fib(4) \implies fib(3) \implies fib(1)

Program Stack

↓Stack↓

frame: main
var: main::f = GARBAGE
frame: fib(4)
var: fib(4)::n = 4
var: fib(4)::nMinus2 = 1
var: fib(4)::nMinus1 = GARBAGE
frame: fib(3)
var: fib(3)::n = 3
var: fib(3)::nMinus2 = 1
frame: fib(1)

↑Heap↑

```
1 /** The fibonacci sequence stack. */
2 #include <iostream>
3 using namespace std;
4 int fib( int n )
5 {
6     if( n <= 2 ) {
7         return 1;           // <==
8     }
9     int nMinus2( fib(n-2) );
10    int nMinus1( fib(n-1) );
11    int r( nMinus2 + nMinus1 );
12    return r;
13 }
14 int main()
15 {
16     int f;
17     f = fib(4);
18     cout << "fibonacci (4)="
19          << f << endl;
20     return 0;
21 }
```

Line 7: Yes, n is 1, return!

discard call frame, return to caller

<OS> \implies main \implies fib(4) \implies fib(3) \implies fib(1)

Program Stack

↓Stack↓

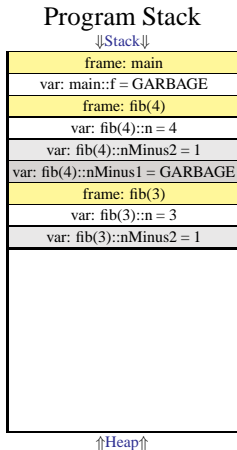
frame: main
var: main::f = GARBAGE
frame: fib(4)
var: fib(4)::n = 4
var: fib(4)::nMinus2 = 1
var: fib(4)::nMinus1 = GARBAGE
frame: fib(3)
var: fib(3)::n = 3
var: fib(3)::nMinus2 = 1

↑Heap↑

```
1 /** The fibonacci sequence stack. */
2 #include <iostream>
3 using namespace std;
4 int fib( int n )
5 {
6     if( n <= 2 ) {
7         return 1;           // <==
8     }
9     int nMinus2( fib(n-2) );
10    int nMinus1( fib(n-1) );
11    int r( nMinus2 + nMinus1 );
12    return r;
13 }
14 int main()
15 {
16     int f;
17     f = fib(4);
18     cout << "fibonacci (4)="
19          << f << endl;
20     return 0;
21 }
```

Line 9: Return from fib(1)

<OS> \implies main \implies fib(4) \implies fib(3)



```
1  /** The fibonacci sequence stack. */
2  #include <iostream>
3  using namespace std;
4  int fib( int n )
5  {
6      if( n <= 2 ) {
7          return 1;
8      }
9      int nMinus2( fib(n-2) );    // <==
10     int nMinus1( fib(n-1) );
11     int r( nMinus2 + nMinus1 );
12     return r;
13 }
14 int main()
15 {
16     int f;
17     f = fib(4);
18     cout << "fibonacci(4)="
19          << f << endl;
20     return 0;
21 }
```

Line 10: Declare nMinus1...

<OS> \implies main \implies fib(4) \implies fib(3)

Program Stack

↓Stack↓

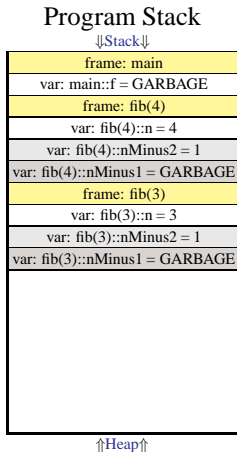
frame: main
var: main::f = GARBAGE
frame: fib(4)
var: fib(4)::n = 4
var: fib(4)::nMinus2 = 1
var: fib(4)::nMinus1 = GARBAGE
frame: fib(3)
var: fib(3)::n = 3
var: fib(3)::nMinus2 = 1
var: fib(3)::nMinus1 = GARBAGE

↑Heap↑

```
1  /** The fibonacci sequence stack. */
2  #include <iostream>
3  using namespace std;
4  int fib( int n )
5  {
6      if( n <= 2 ) {
7          return 1;
8      }
9      int nMinus2( fib(n-2) );
10     int nMinus1( fib(n-1) );    // <==
11     int r( nMinus2 + nMinus1 );
12     return r;
13 }
14 int main()
15 {
16     int f;
17     f = fib(4);
18     cout << "fibonacci(4)="
19          << f << endl;
20     return 0;
21 }
```

Line 10: Call fib(2)

<OS> \implies main \implies fib(4) \implies fib(3)

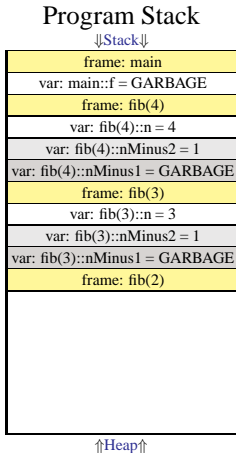


```
1  /** The fibonacci sequence stack. */
2  #include <iostream>
3  using namespace std;
4  int fib( int n )
5  {
6      if( n <= 2 ) {
7          return 1;
8      }
9      int nMinus2( fib(n-2) );
10     int nMinus1( fib(n-1) );    // <==
11     int r( nMinus2 + nMinus1 );
12     return r;
13 }
14 int main()
15 {
16     int f;
17     f = fib(4);
18     cout << "fibonacci(4)="
19          << f << endl;
20     return 0;
21 }
```

Line 10: Call fib(2)

create call frame for fib(2)

<OS> \implies main \implies fib(4) \implies fib(3)

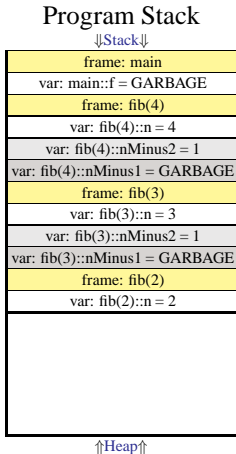


```
1  /** The fibonacci sequence stack. */
2  #include <iostream>
3  using namespace std;
4  int fib( int n )
5  {
6      if( n <= 2 ) {
7          return 1;
8      }
9      int nMinus2( fib(n-2) );
10     int nMinus1( fib(n-1) );    // <==
11     int r( nMinus2 + nMinus1 );
12     return r;
13 }
14 int main()
15 {
16     int f;
17     f = fib(4);
18     cout << "fibonacci(4)="
19          << f << endl;
20     return 0;
21 }
```

Line 10: Call fib(2)

create formal param n of fib(2)

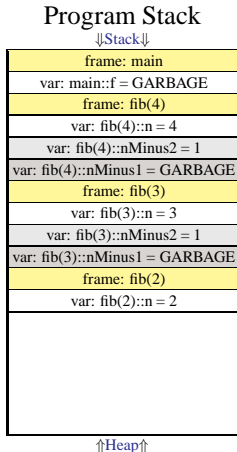
<OS> \implies main \implies fib(4) \implies fib(3)



```
1  /** The fibonacci sequence stack. */
2  #include <iostream>
3  using namespace std;
4  int fib( int n )
5  {
6      if( n <= 2 ) {
7          return 1;
8      }
9      int nMinus2( fib(n-2) );
10     int nMinus1( fib(n-1) );    // <==
11     int r( nMinus2 + nMinus1 );
12     return r;
13 }
14 int main()
15 {
16     int f;
17     f = fib(4);
18     cout << "fibonacci(4)="
19          << f << endl;
20     return 0;
21 }
```

Line 4: Function fib(2) entry

<OS> \implies main \implies fib(4) \implies fib(3) \implies fib(2)



```
1 /** The fibonacci sequence stack. */
2 #include <iostream>
3 using namespace std;
4 int fib( int n )           // <==
5 {
6     if( n <= 2 ) {
7         return 1;
8     }
9     int nMinus2( fib(n-2) );
10    int nMinus1( fib(n-1) );
11    int r( nMinus2 + nMinus1 );
12    return r;
13 }
14 int main()
15 {
16     int f;
17     f = fib(4);
18     cout << "fibonacci (4)="
19          << f << endl;
20     return 0;
21 }
```

Line 6: Is n==1 or n==2?

<OS> \implies main \implies fib(4) \implies fib(3) \implies fib(2)

Program Stack

↓Stack↓

frame: main
var: main::f = GARBAGE
frame: fib(4)
var: fib(4)::n = 4
var: fib(4)::nMinus2 = 1
var: fib(4)::nMinus1 = GARBAGE
frame: fib(3)
var: fib(3)::n = 3
var: fib(3)::nMinus2 = 1
var: fib(3)::nMinus1 = GARBAGE
frame: fib(2)
var: fib(2)::n = 2

↑Heap↑

```
1  /** The fibonacci sequence stack. */
2  #include <iostream>
3  using namespace std;
4  int fib( int n )
5  {
6      if( n <= 2 ) {                // <==
7          return 1;
8      }
9      int nMinus2( fib(n-2) );
10     int nMinus1( fib(n-1) );
11     int r( nMinus2 + nMinus1 );
12     return r;
13 }
14 int main()
15 {
16     int f;
17     f = fib(4);
18     cout << "fibonacci(4)="
19          << f << endl;
20     return 0;
21 }
```

Line 7: Yes, n is 2, return!

<OS> \implies main \implies fib(4) \implies fib(3) \implies fib(2)

Program Stack

↓Stack↓

frame: main
var: main::f = GARBAGE
frame: fib(4)
var: fib(4)::n = 4
var: fib(4)::nMinus2 = 1
var: fib(4)::nMinus1 = GARBAGE
frame: fib(3)
var: fib(3)::n = 3
var: fib(3)::nMinus2 = 1
var: fib(3)::nMinus1 = GARBAGE
frame: fib(2)
var: fib(2)::n = 2

↑Heap↑

```
1 /** The fibonacci sequence stack. */
2 #include <iostream>
3 using namespace std;
4 int fib( int n )
5 {
6     if( n <= 2 ) {
7         return 1;           // <==
8     }
9     int nMinus2( fib(n-2) );
10    int nMinus1( fib(n-1) );
11    int r( nMinus2 + nMinus1 );
12    return r;
13 }
14 int main()
15 {
16     int f;
17     f = fib(4);
18     cout << "fibonacci(4)="
19          << f << endl;
20     return 0;
21 }
```

Line 7: Yes, n is 2, return!

copy return value

<OS> \implies main \implies fib(4) \implies fib(3) \implies fib(2)

Program Stack

↓Stack↓

frame: main
var: main::f = GARBAGE
frame: fib(4)
var: fib(4)::n = 4
var: fib(4)::nMinus2 = 1
var: fib(4)::nMinus1 = GARBAGE
frame: fib(3)
var: fib(3)::n = 3
var: fib(3)::nMinus2 = 1
var: fib(3)::nMinus1 = 1
frame: fib(2)
var: fib(2)::n = 2

↑Heap↑

```
1 /** The fibonacci sequence stack. */
2 #include <iostream>
3 using namespace std;
4 int fib( int n )
5 {
6     if( n <= 2 ) {
7         return 1;           // <==
8     }
9     int nMinus2( fib(n-2) );
10    int nMinus1( fib(n-1) );
11    int r( nMinus2 + nMinus1 );
12    return r;
13 }
14 int main()
15 {
16     int f;
17     f = fib(4);
18     cout << "fibonacci (4)="
19          << f << endl;
20     return 0;
21 }
```

Line 7: Yes, n is 2, return!

discard fib(2) parameter n

<OS> \implies main \implies fib(4) \implies fib(3) \implies fib(2)

Program Stack

↓Stack↓

frame: main
var: main::f = GARBAGE
frame: fib(4)
var: fib(4)::n = 4
var: fib(4)::nMinus2 = 1
var: fib(4)::nMinus1 = GARBAGE
frame: fib(3)
var: fib(3)::n = 3
var: fib(3)::nMinus2 = 1
var: fib(3)::nMinus1 = 1
frame: fib(2)

↑Heap↑

```
1 /** The fibonacci sequence stack. */
2 #include <iostream>
3 using namespace std;
4 int fib( int n )
5 {
6     if ( n <= 2 ) {
7         return 1;           // <==
8     }
9     int nMinus2( fib(n-2) );
10    int nMinus1( fib(n-1) );
11    int r( nMinus2 + nMinus1 );
12    return r;
13 }
14 int main()
15 {
16     int f;
17     f = fib(4);
18     cout << "fibonacci (4)="
19         << f << endl;
20     return 0;
21 }
```

Line 7: Yes, n is 2, return!

discard call frame, return to caller

<OS> \implies main \implies fib(4) \implies fib(3) \implies fib(2)

Program Stack

↓Stack↓

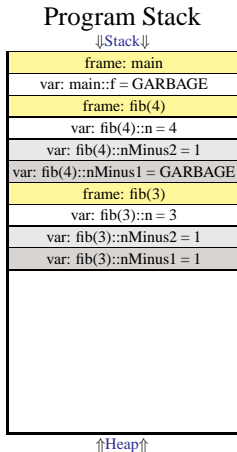
frame: main
var: main::f = GARBAGE
frame: fib(4)
var: fib(4)::n = 4
var: fib(4)::nMinus2 = 1
var: fib(4)::nMinus1 = GARBAGE
frame: fib(3)
var: fib(3)::n = 3
var: fib(3)::nMinus2 = 1
var: fib(3)::nMinus1 = 1

↑Heap↑

```
1 /** The fibonacci sequence stack. */
2 #include <iostream>
3 using namespace std;
4 int fib( int n )
5 {
6     if( n <= 2 ) {
7         return 1; // <==
8     }
9     int nMinus2( fib(n-2) );
10    int nMinus1( fib(n-1) );
11    int r( nMinus2 + nMinus1 );
12    return r;
13 }
14 int main()
15 {
16     int f;
17     f = fib(4);
18     cout << "fibonacci (4)="
19         << f << endl;
20     return 0;
21 }
```

Line 10: Return from fib(2)

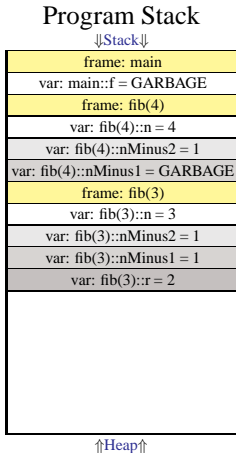
<OS> \implies main \implies fib(4) \implies fib(3)



```
1  /** The fibonacci sequence stack. */
2  #include <iostream>
3  using namespace std;
4  int fib( int n )
5  {
6      if( n <= 2 ) {
7          return 1;
8      }
9      int nMinus2( fib(n-2) );
10     int nMinus1( fib(n-1) );    // <==
11     int r( nMinus2 + nMinus1 );
12     return r;
13 }
14 int main()
15 {
16     int f;
17     f = fib(4);
18     cout << "fibonacci(4)="
19          << f << endl;
20     return 0;
21 }
```

Line 11: Declare r(nMinus2+nMinus1)

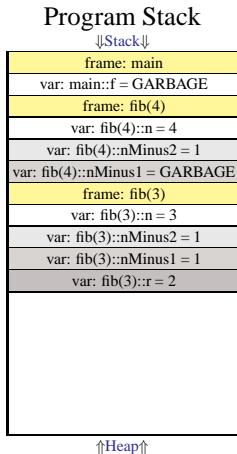
<OS> \implies main \implies fib(4) \implies fib(3)



```
1  /** The fibonacci sequence stack. */
2  #include <iostream>
3  using namespace std;
4  int fib( int n )
5  {
6      if( n <= 2 ) {
7          return 1;
8      }
9      int nMinus2( fib(n-2) );
10     int nMinus1( fib(n-1) );
11     int r( nMinus2 + nMinus1 ); // <==
12     return r;
13 }
14 int main()
15 {
16     int f;
17     f = fib(4);
18     cout << "fibonacci(4)="
19          << f << endl;
20     return 0;
21 }
```

Line 12: Return r=2 from fib(3)

<OS> \implies main \implies fib(4) \implies fib(3)

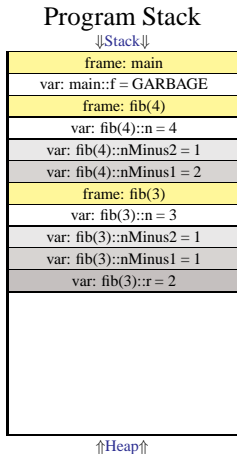


```
1  /** The fibonacci sequence stack. */
2  #include <iostream>
3  using namespace std;
4  int fib( int n )
5  {
6      if( n <= 2 ) {
7          return 1;
8      }
9      int nMinus2( fib(n-2) );
10     int nMinus1( fib(n-1) );
11     int r( nMinus2 + nMinus1 );
12     return r; // <==
13 }
14 int main()
15 {
16     int f;
17     f = fib(4);
18     cout << "fibonacci(4)="
19          << f << endl;
20     return 0;
21 }
```

Line 12: Return r=2 from fib(3)

copy return value

<OS> \implies main \implies fib(4) \implies fib(3)

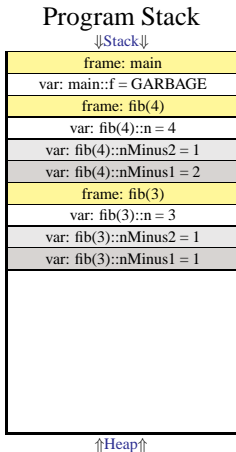


```
1  /** The fibonacci sequence stack. */
2  #include <iostream>
3  using namespace std;
4  int fib( int n )
5  {
6      if( n <= 2 ) {
7          return 1;
8      }
9      int nMinus2( fib(n-2) );
10     int nMinus1( fib(n-1) );
11     int r( nMinus2 + nMinus1 );
12     return r; // <==
13 }
14 int main()
15 {
16     int f;
17     f = fib(4);
18     cout << "fibonacci(4)="
19          << f << endl;
20     return 0;
21 }
```

Line 12: Return r=2 from fib(3)

discard the auto (local) variable fib(3)::r

<OS> \implies main \implies fib(4) \implies fib(3)

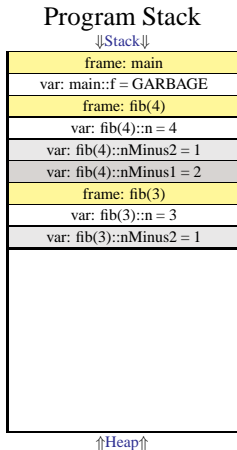


```
1  /** The fibonacci sequence stack. */
2  #include <iostream>
3  using namespace std;
4  int fib( int n )
5  {
6      if( n <= 2 ) {
7          return 1;
8      }
9      int nMinus2( fib(n-2) );
10     int nMinus1( fib(n-1) );
11     int r( nMinus2 + nMinus1 );
12     return r; // <==
13 }
14 int main()
15 {
16     int f;
17     f = fib(4);
18     cout << "fibonacci(4)="
19          << f << endl;
20     return 0;
21 }
```

Line 12: Return r=2 from fib(3)

discard the auto (local) variable fib(3)::nMinus1

<OS> \implies main \implies fib(4) \implies fib(3)

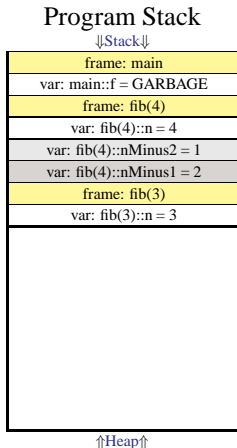


```
1  /** The fibonacci sequence stack. */
2  #include <iostream>
3  using namespace std;
4  int fib( int n )
5  {
6      if( n <= 2 ) {
7          return 1;
8      }
9      int nMinus2( fib( n-2 ) );
10     int nMinus1( fib( n-1 ) );
11     int r( nMinus2 + nMinus1 );
12     return r; // <==
13 }
14 int main()
15 {
16     int f;
17     f = fib(4);
18     cout << "fibonacci (4)="
19          << f << endl;
20     return 0;
21 }
```

Line 12: Return r=2 from fib(3)

discard the auto (local) variable fib(3)::nMinus2

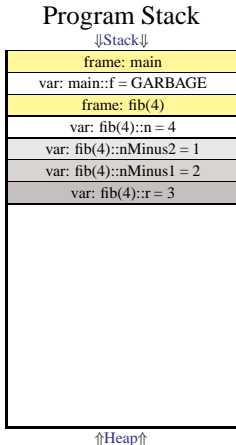
<OS> \implies main \implies fib(4) \implies fib(3)



```
1  /** The fibonacci sequence stack. */
2  #include <iostream>
3  using namespace std;
4  int fib( int n )
5  {
6      if( n <= 2 ) {
7          return 1;
8      }
9      int nMinus2( fib(n-2) );
10     int nMinus1( fib(n-1) );
11     int r( nMinus2 + nMinus1 );
12     return r; // <==
13 }
14 int main()
15 {
16     int f;
17     f = fib(4);
18     cout << "fibonacci(4)="
19          << f << endl;
20     return 0;
21 }
```


Line 11: Declare r(nMinus2+nMinus1)

<OS> \implies main \implies fib(4)

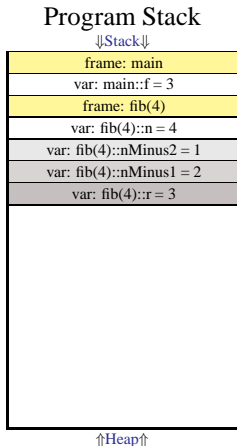


```
1  /** The fibonacci sequence stack. */
2  #include <iostream>
3  using namespace std;
4  int fib( int n )
5  {
6      if( n <= 2 ) {
7          return 1;
8      }
9      int nMinus2( fib(n-2) );
10     int nMinus1( fib(n-1) );
11     int r( nMinus2 + nMinus1 ); // <==
12     return r;
13 }
14 int main()
15 {
16     int f;
17     f = fib(4);
18     cout << "fibonacci(4)="
19          << f << endl;
20     return 0;
21 }
```


Line 12: Return r=3 from fib(4)

copy return value

<OS> \implies main \implies fib(4)

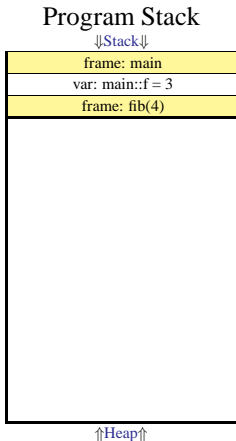


```
1  /** The fibonacci sequence stack. */
2  #include <iostream>
3  using namespace std;
4  int fib( int n )
5  {
6      if( n <= 2 ) {
7          return 1;
8      }
9      int nMinus2( fib(n-2) );
10     int nMinus1( fib(n-1) );
11     int r( nMinus2 + nMinus1 );
12     return r;           // <==
13 }
14 int main()
15 {
16     int f;
17     f = fib(4);
18     cout << "fibonacci(4)="
19          << f << endl;
20     return 0;
21 }
```


Line 12: Return r=3 from fib(4)

discard fib(4) parameter n

<OS> \implies main \implies fib(4)



```
1  /** The fibonacci sequence stack. */
2  #include <iostream>
3  using namespace std;
4  int fib( int n )
5  {
6      if( n <= 2 ) {
7          return 1;
8      }
9      int nMinus2( fib(n-2) );
10     int nMinus1( fib(n-1) );
11     int r( nMinus2 + nMinus1 );
12     return r;           // <==
13 }
14 int main()
15 {
16     int f;
17     f = fib(4);
18     cout << "fibonacci(4)="
19          << f << endl;
20     return 0;
21 }
```


Line 20: Return to OS

discard the auto (local) variable `main::f`

<OS> \implies main



```
1 /** The fibonacci sequence stack. */
2 #include <iostream>
3 using namespace std;
4 int fib( int n )
5 {
6     if( n <= 2 ) {
7         return 1;
8     }
9     int nMinus2( fib(n-2) );
10    int nMinus1( fib(n-1) );
11    int r( nMinus2 + nMinus1 );
12    return r;
13 }
14 int main()
15 {
16     int f;
17     f = fib(4);
18     cout << "fibonacci (4)="
19         << f << endl;
20     return 0; // <==
21 }
```

Line 20: Return to OS

discard call frame, return to caller

<OS> \implies main



```
1  /** The fibonacci sequence stack. */
2  #include <iostream>
3  using namespace std;
4  int fib( int n )
5  {
6      if( n <= 2 ) {
7          return 1;
8      }
9      int nMinus2( fib(n-2) );
10     int nMinus1( fib(n-1) );
11     int r( nMinus2 + nMinus1 );
12     return r;
13 }
14 int main()
15 {
16     int f;
17     f = fib(4);
18     cout << "fibonacci (4)="
19          << f << endl;
20     return 0; // <==
21 }
```

Return from main

<OS>



```
1 /** The fibonacci sequence stack. */
2 #include <iostream>
3 using namespace std;
4 int fib( int n )
5 {
6     if( n <= 2 ) {
7         return 1;
8     }
9     int nMinus2( fib(n-2) );
10    int nMinus1( fib(n-1) );
11    int r( nMinus2 + nMinus1 );
12    return r;
13 }
14 int main()
15 {
16     int f;
17     f = fib(4);
18     cout << "fibonacci (4)="
19          << f << endl;
20     return 0;
21 }
```

finis