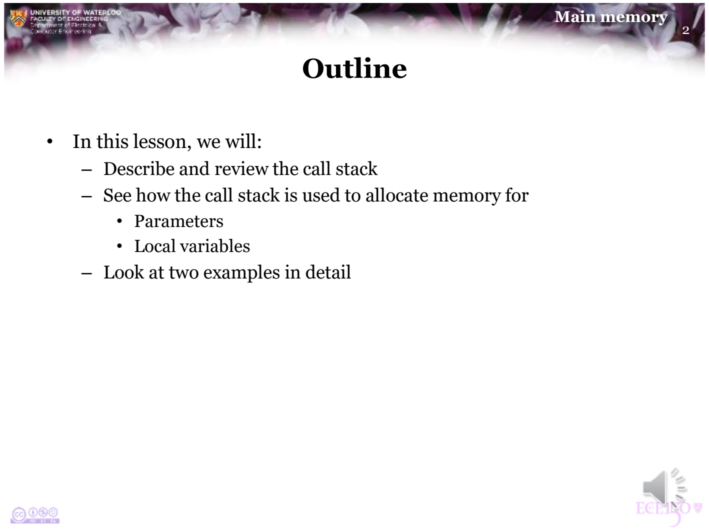
UNIVERSITY OF WATERLOO
FACULTY OF ENGINEERING
Department of Electrical & Computer Engineering

ECE 150 *Fundamentals of Programming*

The call stack

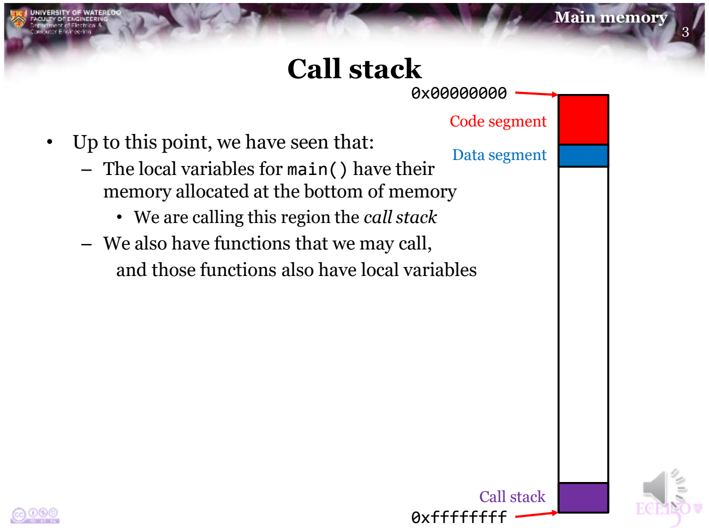
Prof. Hiren Patel, Ph.D.
Prof. Werner Dietl, Ph. D.
Douglas Wilhelm Harder, M.Math. LEL



Main memory 2

Outline

- In this lesson, we will:
  - Describe and review the call stack
  - See how the call stack is used to allocate memory for
    - Parameters
    - Local variables
  - Look at two examples in detail

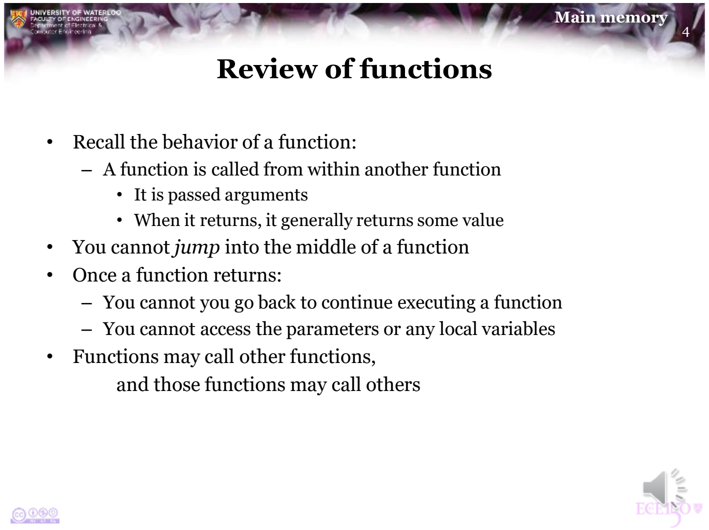


Main memory 3

Call stack

- Up to this point, we have seen that:
  - The local variables for `main()` have their memory allocated at the bottom of memory
    - We are calling this region the *call stack*
  - We also have functions that we may call, and those functions also have local variables

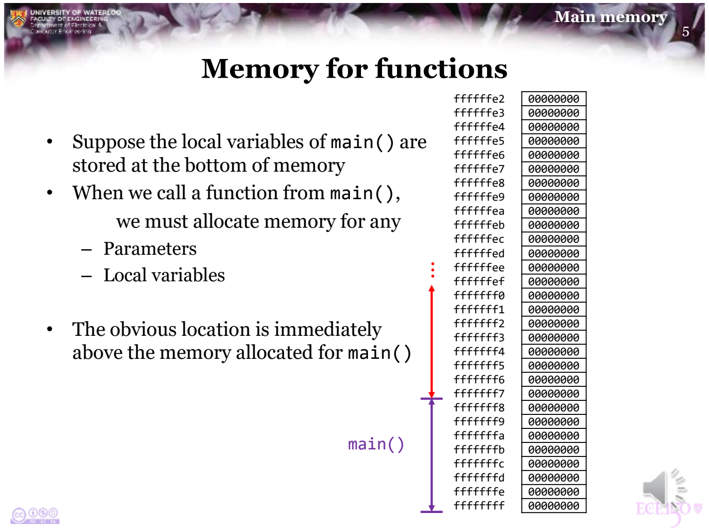
0x00000000
Code segment
Data segment
Call stack
0xffffffff



Main memory 4

Review of functions

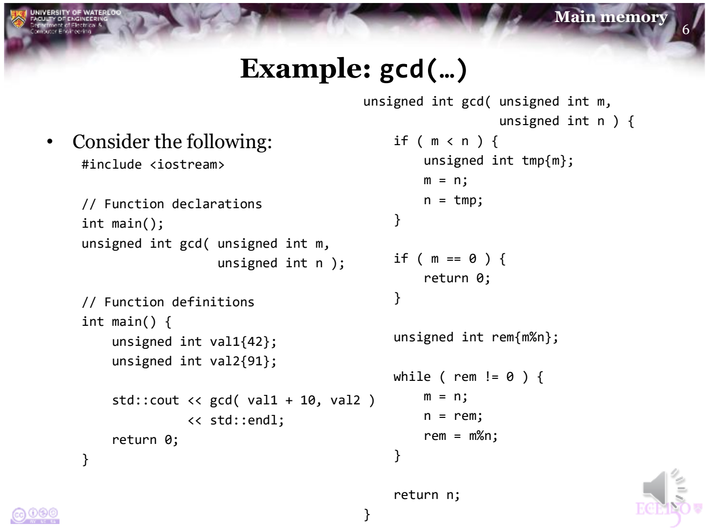
- Recall the behavior of a function:
  - A function is called from within another function
    - It is passed arguments
    - When it returns, it generally returns some value
- You cannot *jump* into the middle of a function
- Once a function returns:
  - You cannot you go back to continue executing a function
  - You cannot access the parameters or any local variables
- Functions may call other functions, and those functions may call others



Memory for functions

- Suppose the local variables of `main()` are stored at the bottom of memory
- When we call a function from `main()`,
  we must allocate memory for any
  - Parameters
  - Local variables
- The obvious location is immediately above the memory allocated for `main()`

```
ffffffe2  00000000
ffffffe3  00000000
ffffffe4  00000000
ffffffe5  00000000
ffffffe6  00000000
ffffffe7  00000000
ffffffe8  00000000
ffffffe9  00000000
ffffffea  00000000
ffffffeb  00000000
ffffffec  00000000
ffffffed  00000000
ffffffee  00000000
ffffffef  00000000
fffffff0  00000000
fffffff1  00000000
fffffff2  00000000
fffffff3  00000000
fffffff4  00000000
fffffff5  00000000
fffffff6  00000000
fffffff7  00000000
fffffff8  00000000
fffffff9  00000000
fffffffa  00000000
fffffffb  00000000
fffffffc  00000000
fffffffd  00000000
fffffffe  00000000
ffffffff  00000000
```

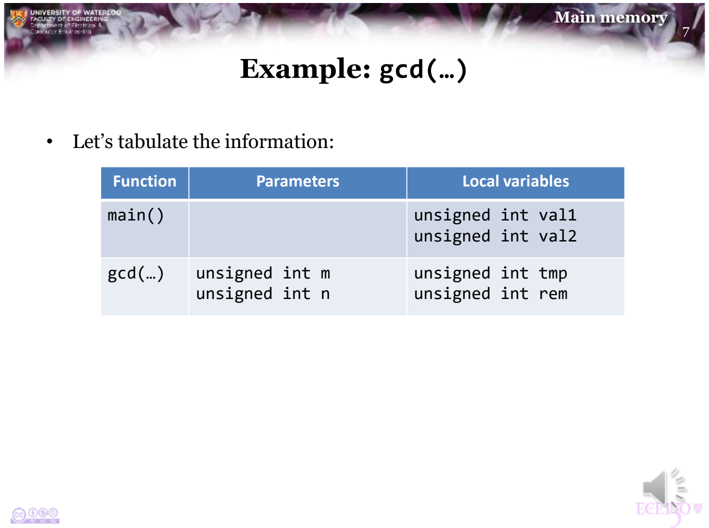


Example: gcd(…)

- Consider the following:

```
#include <iostream>

// Function declarations
int main();
unsigned int gcd( unsigned int m,
                  unsigned int n );

// Function definitions
int main() {
    unsigned int val1{42};
    unsigned int val2{91};

    std::cout << gcd( val1 + 10, val2 )
              << std::endl;
    return 0;
}
```

```
unsigned int gcd( unsigned int m,
                  unsigned int n ) {
    if ( m < n ) {
        unsigned int tmp{m};
        m = n;
        n = tmp;
    }

    if ( m == 0 ) {
        return 0;
    }

    unsigned int rem{m%n};

    while ( rem != 0 ) {
        m = n;
        n = rem;
        rem = m%n;
    }

    return n;
}
```



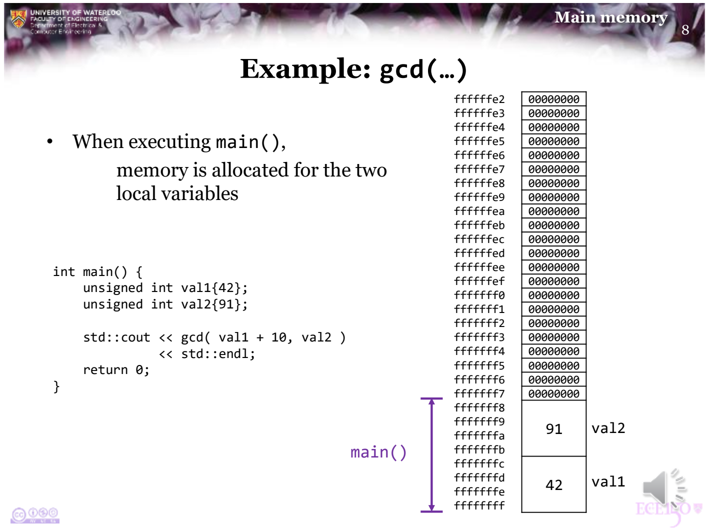
Example: gcd(…)

- Let's tabulate the information:

Function	Parameters	Local variables

`main()`		`unsigned int val1`
		`unsigned int val2`

`gcd(…)`	`unsigned int m`
	`unsigned int n`
		`unsigned int tmp`
		`unsigned int rem`

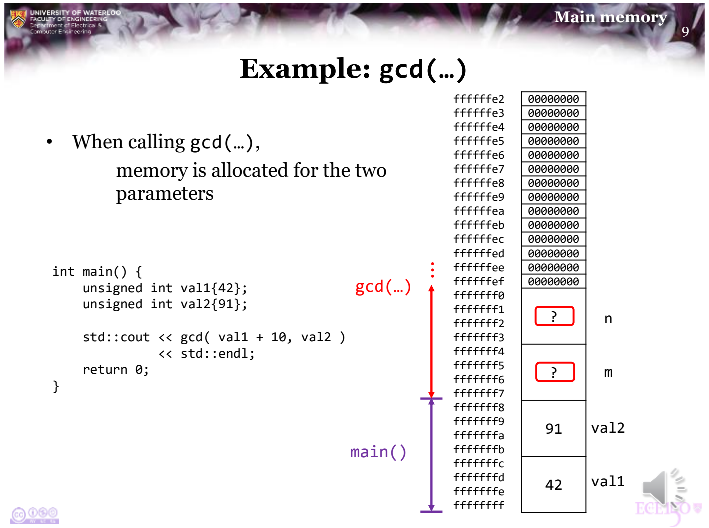


Example: gcd(…)

- When executing `main()`,
  memory is allocated for the two local variables

```
int main() {
    unsigned int val1{42};
    unsigned int val2{91};

    std::cout << gcd( val1 + 10, val2 )
              << std::endl;
    return 0;
}
```

```
ffffffe2  00000000
ffffffe3  00000000
ffffffe4  00000000
ffffffe5  00000000
ffffffe6  00000000
ffffffe7  00000000
ffffffe8  00000000
ffffffe9  00000000
ffffffea  00000000
ffffffeb  00000000
ffffffec  00000000
ffffffed  00000000
ffffffee  00000000
ffffffef  00000000
fffffff0  00000000
fffffff1  00000000
fffffff2  00000000
fffffff3  00000000
fffffff4  00000000
fffffff5  00000000
fffffff6  00000000
fffffff7  00000000
fffffff8
fffffff9     91      val2
fffffffa
fffffffb
fffffffc
fffffffd     42      val1
fffffffe
ffffffff
```

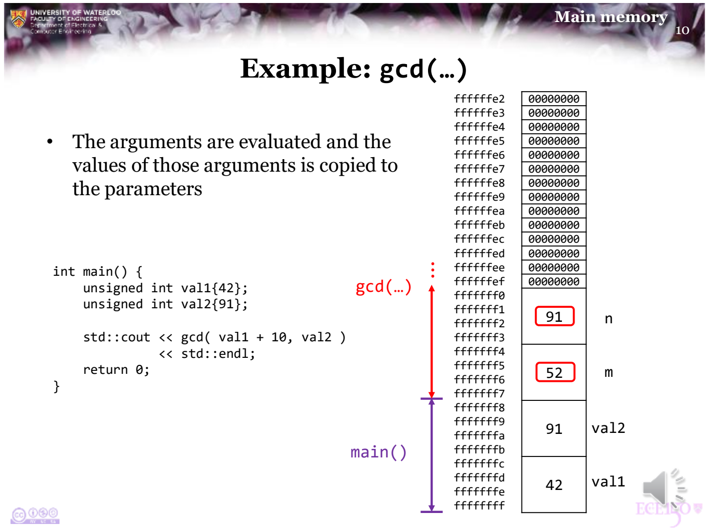


Example: gcd(…)

- When calling gcd(…),
  memory is allocated for the two parameters

```
int main() {
    unsigned int val1{42};
    unsigned int val2{91};

    std::cout << gcd( val1 + 10, val2 )
              << std::endl;
    return 0;
}
```

ffffffe2	00000000	
ffffffe3	00000000	
ffffffe4	00000000	
ffffffe5	00000000	
ffffffe6	00000000	
ffffffe7	00000000	
ffffffe8	00000000	
ffffffe9	00000000	
ffffffea	00000000	
ffffffeb	00000000	
ffffffec	00000000	
ffffffed	00000000	
ffffffee	00000000	
ffffffef	00000000	
fffffff0	?	n
fffffff1		
fffffff2		
fffffff3		
fffffff4	?	m
fffffff5		
fffffff6		
fffffff7		
fffffff8	91	val2
fffffff9		
fffffffa		
fffffffb		
fffffffc	42	val1
fffffffd		
fffffffe		
ffffffff		

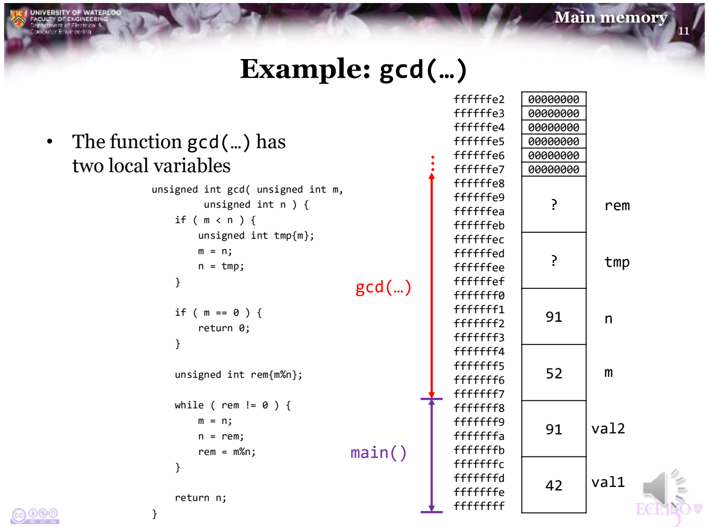


Example: gcd(…)

- The arguments are evaluated and the values of those arguments is copied to the parameters

```
int main() {
    unsigned int val1{42};
    unsigned int val2{91};

    std::cout << gcd( val1 + 10, val2 )
              << std::endl;
    return 0;
}
```

ffffffe2	00000000	
ffffffe3	00000000	
ffffffe4	00000000	
ffffffe5	00000000	
ffffffe6	00000000	
ffffffe7	00000000	
ffffffe8	00000000	
ffffffe9	00000000	
ffffffea	00000000	
ffffffeb	00000000	
ffffffec	00000000	
ffffffed	00000000	
ffffffee	00000000	
ffffffef	00000000	
fffffff0	91	n
fffffff1		
fffffff2		
fffffff3		
fffffff4	52	m
fffffff5		
fffffff6		
fffffff7		
fffffff8	91	val2
fffffff9		
fffffffa		
fffffffb		
fffffffc	42	val1
fffffffd		
fffffffe		
ffffffff		

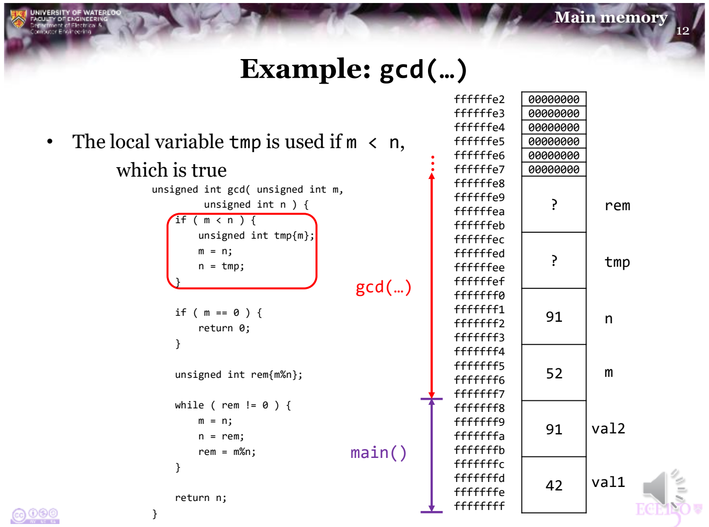


Example: gcd(…)

- The function gcd(…) has two local variables

```
unsigned int gcd( unsigned int m,
                  unsigned int n ) {
    if ( m < n ) {
        unsigned int tmp{m};
        m = n;
        n = tmp;
    }

    if ( m == 0 ) {
        return 0;
    }

    unsigned int rem{m%n};

    while ( rem != 0 ) {
        m = n;
        n = rem;
        rem = m%n;
    }

    return n;
}
```

ffffffe2	00000000	
ffffffe3	00000000	
ffffffe4	00000000	
ffffffe5	00000000	
ffffffe6	00000000	
ffffffe7	00000000	
ffffffe8	?	rem
ffffffe9		
ffffffea		
ffffffeb		
ffffffec	?	tmp
ffffffed		
ffffffee		
ffffffef		
fffffff0	91	n
fffffff1		
fffffff2		
fffffff3		
fffffff4	52	m
fffffff5		
fffffff6		
fffffff7		
fffffff8	91	val2
fffffff9		
fffffffa		
fffffffb		
fffffffc	42	val1
fffffffd		
fffffffe		
ffffffff		

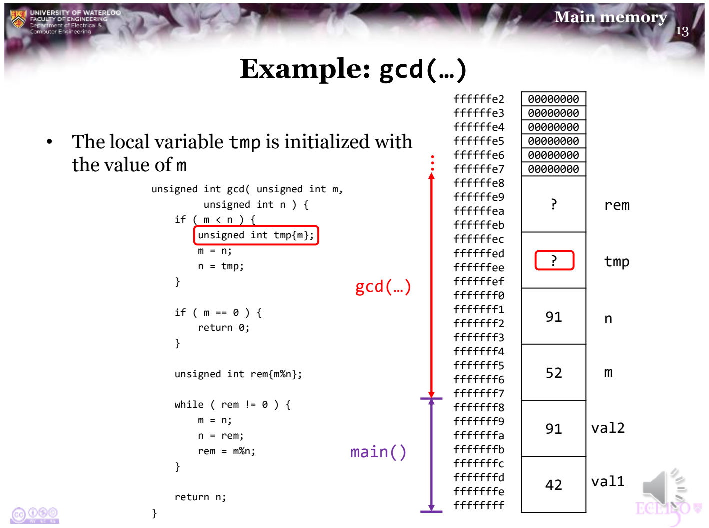


Example: gcd(…)

- The local variable tmp is used if m < n,
  which is true

```
unsigned int gcd( unsigned int m,
                  unsigned int n ) {
    if ( m < n ) {
        unsigned int tmp{m};
        m = n;
        n = tmp;
    }

    if ( m == 0 ) {
        return 0;
    }

    unsigned int rem{m%n};

    while ( rem != 0 ) {
        m = n;
        n = rem;
        rem = m%n;
    }

    return n;
}
```

ffffffe2	00000000	
ffffffe3	00000000	
ffffffe4	00000000	
ffffffe5	00000000	
ffffffe6	00000000	
ffffffe7	00000000	
ffffffe8	?	rem
ffffffe9		
ffffffea		
ffffffeb		
ffffffec	?	tmp
ffffffed		
ffffffee		
ffffffef		
fffffff0	91	n
fffffff1		
fffffff2		
fffffff3		
fffffff4	52	m
fffffff5		
fffffff6		
fffffff7		
fffffff8	91	val2
fffffff9		
fffffffa		
fffffffb		
fffffffc	42	val1
fffffffd		
fffffffe		
ffffffff		

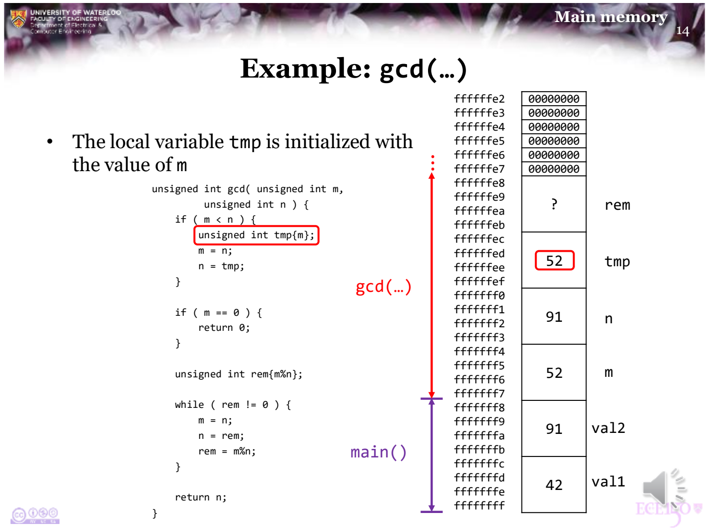


Example: gcd(…)

- The local variable tmp is initialized with the value of m

```
unsigned int gcd( unsigned int m,
                  unsigned int n ) {
    if ( m < n ) {
        unsigned int tmp{m};
        m = n;
        n = tmp;
    }

    if ( m == 0 ) {
        return 0;
    }

    unsigned int rem{m%n};

    while ( rem != 0 ) {
        m = n;
        n = rem;
        rem = m%n;
    }

    return n;
}
```

```
ffffffe2  00000000
ffffffe3  00000000
ffffffe4  00000000
ffffffe5  00000000
ffffffe6  00000000
ffffffe7  00000000
ffffffe8
ffffffe9
ffffffea      ?      rem
ffffffeb
ffffffec
ffffffed      ?      tmp
ffffffee
ffffffef
fffffff0
fffffff1     91      n
fffffff2
fffffff3
fffffff4
fffffff5     52      m
fffffff6
fffffff7
fffffff8
fffffff9     91      val2
fffffffa
fffffffb
fffffffc
fffffffd     42      val1
fffffffe
ffffffff
```

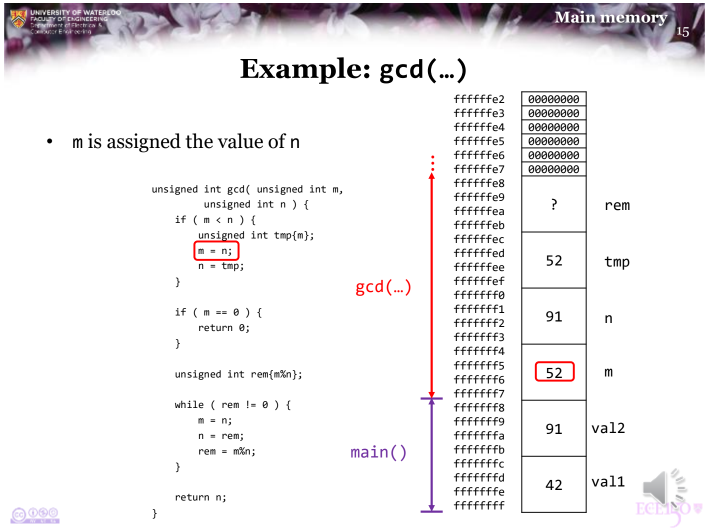


Example: gcd(…)

- The local variable tmp is initialized with the value of m

```
unsigned int gcd( unsigned int m,
                  unsigned int n ) {
    if ( m < n ) {
        unsigned int tmp{m};
        m = n;
        n = tmp;
    }

    if ( m == 0 ) {
        return 0;
    }

    unsigned int rem{m%n};

    while ( rem != 0 ) {
        m = n;
        n = rem;
        rem = m%n;
    }

    return n;
}
```

```
ffffffe2  00000000
ffffffe3  00000000
ffffffe4  00000000
ffffffe5  00000000
ffffffe6  00000000
ffffffe7  00000000
ffffffe8
ffffffe9
ffffffea      ?      rem
ffffffeb
ffffffec
ffffffed     52      tmp
ffffffee
ffffffef
fffffff0
fffffff1     91      n
fffffff2
fffffff3
fffffff4
fffffff5     52      m
fffffff6
fffffff7
fffffff8
fffffff9     91      val2
fffffffa
fffffffb
fffffffc
fffffffd     42      val1
fffffffe
ffffffff
```

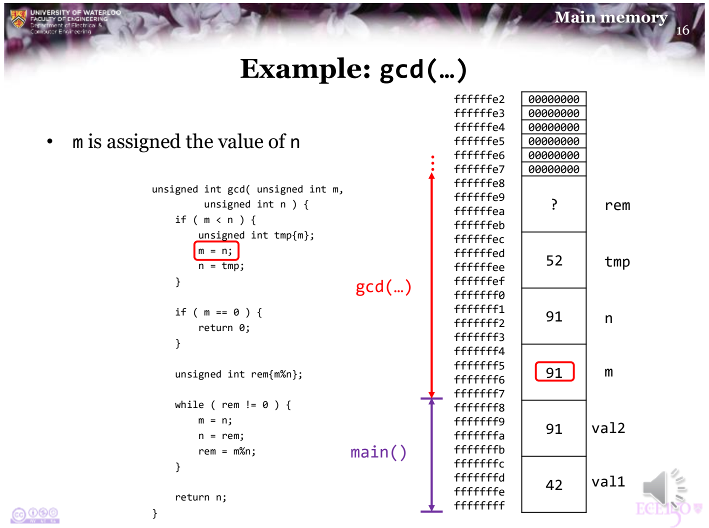


Example: gcd(…)

- m is assigned the value of n

```
unsigned int gcd( unsigned int m,
                  unsigned int n ) {
    if ( m < n ) {
        unsigned int tmp{m};
        m = n;
        n = tmp;
    }

    if ( m == 0 ) {
        return 0;
    }

    unsigned int rem{m%n};

    while ( rem != 0 ) {
        m = n;
        n = rem;
        rem = m%n;
    }

    return n;
}
```

```
ffffffe2  00000000
ffffffe3  00000000
ffffffe4  00000000
ffffffe5  00000000
ffffffe6  00000000
ffffffe7  00000000
ffffffe8
ffffffe9
ffffffea      ?      rem
ffffffeb
ffffffec
ffffffed     52      tmp
ffffffee
ffffffef
fffffff0
fffffff1     91      n
fffffff2
fffffff3
fffffff4
fffffff5     52      m
fffffff6
fffffff7
fffffff8
fffffff9     91      val2
fffffffa
fffffffb
fffffffc
fffffffd     42      val1
fffffffe
ffffffff
```

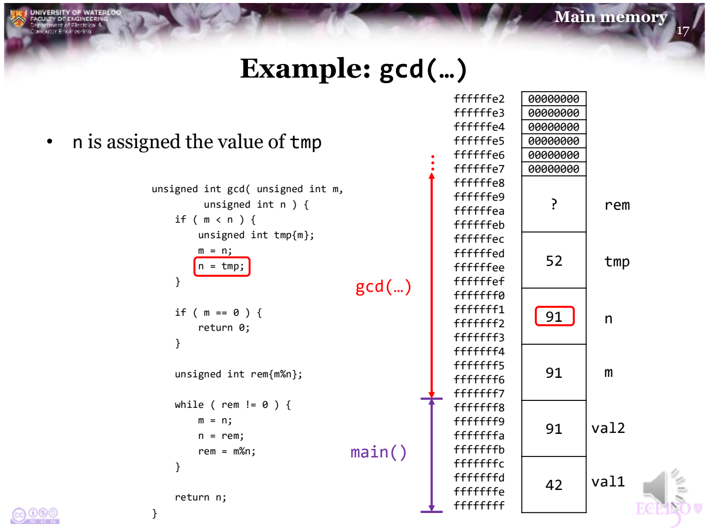


Example: gcd(…)

- m is assigned the value of n

```
unsigned int gcd( unsigned int m,
                  unsigned int n ) {
    if ( m < n ) {
        unsigned int tmp{m};
        m = n;
        n = tmp;
    }

    if ( m == 0 ) {
        return 0;
    }

    unsigned int rem{m%n};

    while ( rem != 0 ) {
        m = n;
        n = rem;
        rem = m%n;
    }

    return n;
}
```

```
ffffffe2  00000000
ffffffe3  00000000
ffffffe4  00000000
ffffffe5  00000000
ffffffe6  00000000
ffffffe7  00000000
ffffffe8
ffffffe9
ffffffea      ?      rem
ffffffeb
ffffffec
ffffffed     52      tmp
ffffffee
ffffffef
fffffff0
fffffff1     91      n
fffffff2
fffffff3
fffffff4
fffffff5     91      m
fffffff6
fffffff7
fffffff8
fffffff9     91      val2
fffffffa
fffffffb
fffffffc
fffffffd     42      val1
fffffffe
ffffffff
```

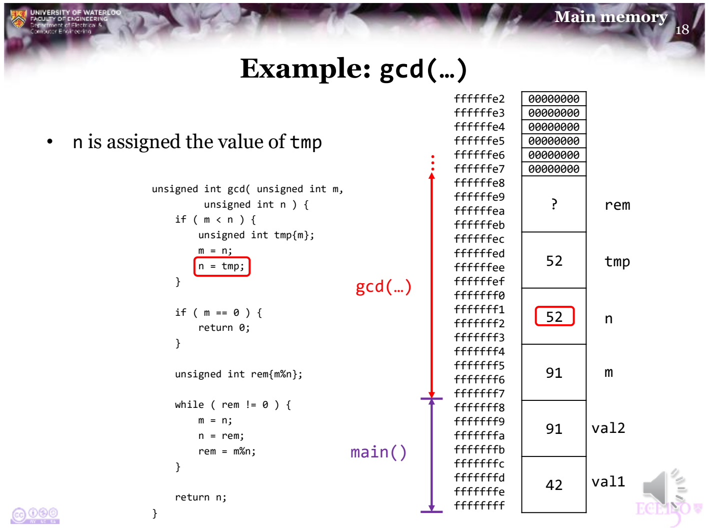


Example: gcd(…)

- n is assigned the value of tmp

```
unsigned int gcd( unsigned int m,
                  unsigned int n ) {
    if ( m < n ) {
        unsigned int tmp{m};
        m = n;
        n = tmp;
    }

    if ( m == 0 ) {
        return 0;
    }

    unsigned int rem{m%n};

    while ( rem != 0 ) {
        m = n;
        n = rem;
        rem = m%n;
    }

    return n;
}
```

ffffffe2	00000000		
ffffffe3	00000000		
ffffffe4	00000000		
ffffffe5	00000000		
ffffffe6	00000000		
ffffffe7	00000000		
ffffffe8	?	rem	gcd(…)
ffffffe9			
ffffffea			
ffffffeb			
ffffffec	52	tmp	
ffffffed			
ffffffee			
ffffffef			
fffffff0	91	n	
fffffff1			
fffffff2			
fffffff3			
fffffff4	91	m	
fffffff5			
fffffff6			
fffffff7			
fffffff8	91	val2	main()
fffffff9			
fffffffa			
fffffffb			
fffffffc	42	val1	
fffffffd			
fffffffe			
ffffffff			

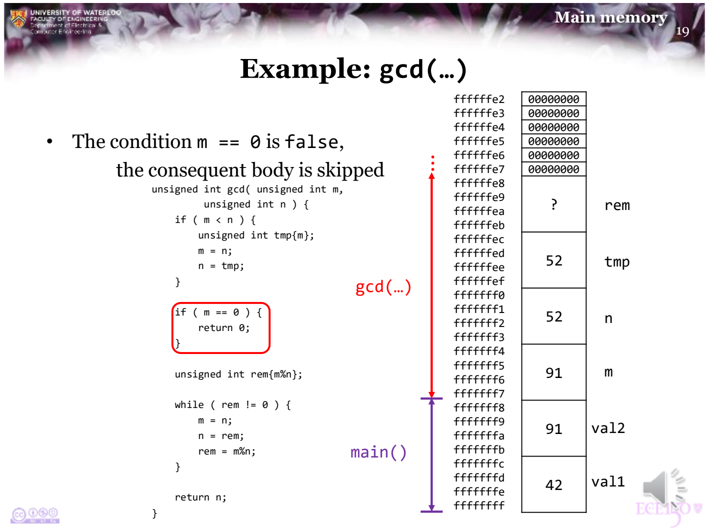


Example: gcd(…)

- n is assigned the value of tmp

```
unsigned int gcd( unsigned int m,
                  unsigned int n ) {
    if ( m < n ) {
        unsigned int tmp{m};
        m = n;
        n = tmp;
    }

    if ( m == 0 ) {
        return 0;
    }

    unsigned int rem{m%n};

    while ( rem != 0 ) {
        m = n;
        n = rem;
        rem = m%n;
    }

    return n;
}
```

ffffffe2	00000000		
ffffffe3	00000000		
ffffffe4	00000000		
ffffffe5	00000000		
ffffffe6	00000000		
ffffffe7	00000000		
ffffffe8	?	rem	gcd(…)
ffffffe9			
ffffffea			
ffffffeb			
ffffffec	52	tmp	
ffffffed			
ffffffee			
ffffffef			
fffffff0	52	n	
fffffff1			
fffffff2			
fffffff3			
fffffff4	91	m	
fffffff5			
fffffff6			
fffffff7			
fffffff8	91	val2	main()
fffffff9			
fffffffa			
fffffffb			
fffffffc	42	val1	
fffffffd			
fffffffe			
ffffffff			

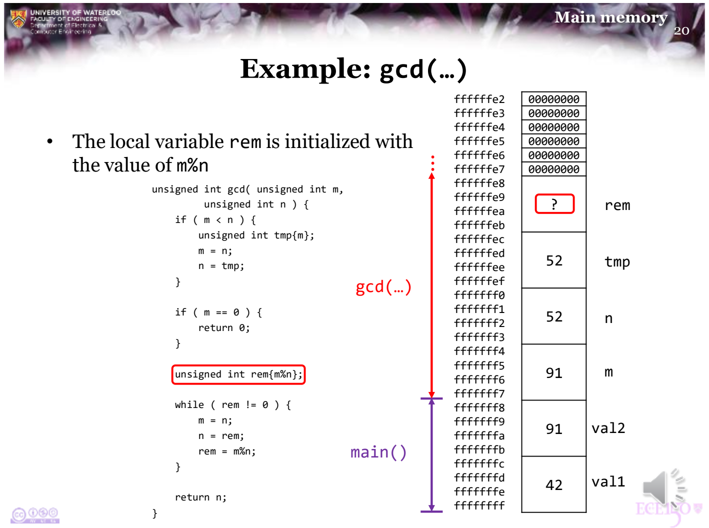


Example: gcd(…)

- The condition m == 0 is false,
  the consequent body is skipped

```
unsigned int gcd( unsigned int m,
                  unsigned int n ) {
    if ( m < n ) {
        unsigned int tmp{m};
        m = n;
        n = tmp;
    }

    if ( m == 0 ) {
        return 0;
    }

    unsigned int rem{m%n};

    while ( rem != 0 ) {
        m = n;
        n = rem;
        rem = m%n;
    }

    return n;
}
```

ffffffe2	00000000		
ffffffe3	00000000		
ffffffe4	00000000		
ffffffe5	00000000		
ffffffe6	00000000		
ffffffe7	00000000		
ffffffe8	?	rem	gcd(…)
ffffffe9			
ffffffea			
ffffffeb			
ffffffec	52	tmp	
ffffffed			
ffffffee			
ffffffef			
fffffff0	52	n	
fffffff1			
fffffff2			
fffffff3			
fffffff4	91	m	
fffffff5			
fffffff6			
fffffff7			
fffffff8	91	val2	main()
fffffff9			
fffffffa			
fffffffb			
fffffffc	42	val1	
fffffffd			
fffffffe			
ffffffff			

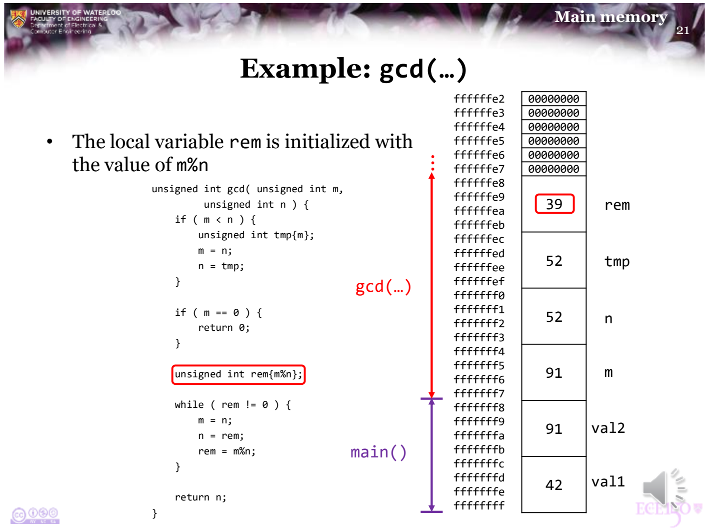


Example: gcd(…)

- The local variable rem is initialized with the value of m%n

```
unsigned int gcd( unsigned int m,
                  unsigned int n ) {
    if ( m < n ) {
        unsigned int tmp{m};
        m = n;
        n = tmp;
    }

    if ( m == 0 ) {
        return 0;
    }

    unsigned int rem{m%n};

    while ( rem != 0 ) {
        m = n;
        n = rem;
        rem = m%n;
    }

    return n;
}
```

ffffffe2	00000000		
ffffffe3	00000000		
ffffffe4	00000000		
ffffffe5	00000000		
ffffffe6	00000000		
ffffffe7	00000000		
ffffffe8	?	rem	gcd(…)
ffffffe9			
ffffffea			
ffffffeb			
ffffffec	52	tmp	
ffffffed			
ffffffee			
ffffffef			
fffffff0	52	n	
fffffff1			
fffffff2			
fffffff3			
fffffff4	91	m	
fffffff5			
fffffff6			
fffffff7			
fffffff8	91	val2	main()
fffffff9			
fffffffa			
fffffffb			
fffffffc	42	val1	
fffffffd			
fffffffe			
ffffffff			



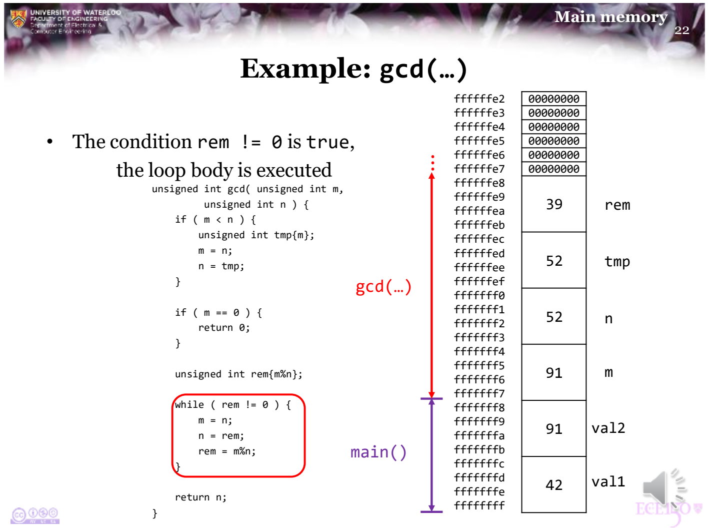
Main memory 21

Example: gcd(…)

- The local variable `rem` is initialized with the value of `m%n`

```
unsigned int gcd( unsigned int m,
         unsigned int n ) {
    if ( m < n ) {
        unsigned int tmp{m};
        m = n;
        n = tmp;
    }

    if ( m == 0 ) {
        return 0;
    }

    unsigned int rem{m%n};

    while ( rem != 0 ) {
        m = n;
        n = rem;
        rem = m%n;
    }

    return n;
}
```

ffffffe2	00000000	
ffffffe3	00000000	
ffffffe4	00000000	
ffffffe5	00000000	
ffffffe6	00000000	
ffffffe7	00000000	
ffffffe8	39	rem
ffffffe9		
ffffffea		
ffffffeb		
ffffffec	52	tmp
ffffffed		
ffffffee		
ffffffef		
fffffff0	52	n
fffffff1		
fffffff2		
fffffff3		
fffffff4	91	m
fffffff5		
fffffff6		
fffffff7		
fffffff8	91	val2
fffffff9		
fffffffa		
fffffffb		
fffffffc	42	val1
fffffffd		
fffffffe		
ffffffff		



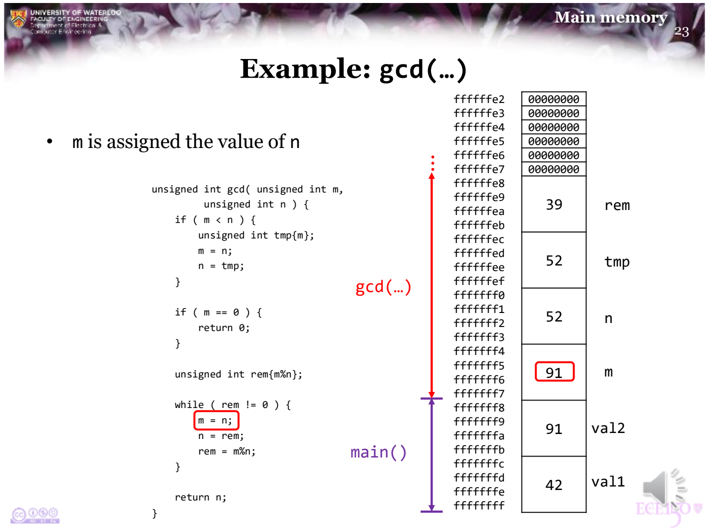
Main memory 22

Example: gcd(…)

- The condition `rem != 0` is `true`, the loop body is executed

```
unsigned int gcd( unsigned int m,
         unsigned int n ) {
    if ( m < n ) {
        unsigned int tmp{m};
        m = n;
        n = tmp;
    }

    if ( m == 0 ) {
        return 0;
    }

    unsigned int rem{m%n};

    while ( rem != 0 ) {
        m = n;
        n = rem;
        rem = m%n;
    }

    return n;
}
```

ffffffe2	00000000	
ffffffe3	00000000	
ffffffe4	00000000	
ffffffe5	00000000	
ffffffe6	00000000	
ffffffe7	00000000	
ffffffe8	39	rem
ffffffe9		
ffffffea		
ffffffeb		
ffffffec	52	tmp
ffffffed		
ffffffee		
ffffffef		
fffffff0	52	n
fffffff1		
fffffff2		
fffffff3		
fffffff4	91	m
fffffff5		
fffffff6		
fffffff7		
fffffff8	91	val2
fffffff9		
fffffffa		
fffffffb		
fffffffc	42	val1
fffffffd		
fffffffe		
ffffffff		



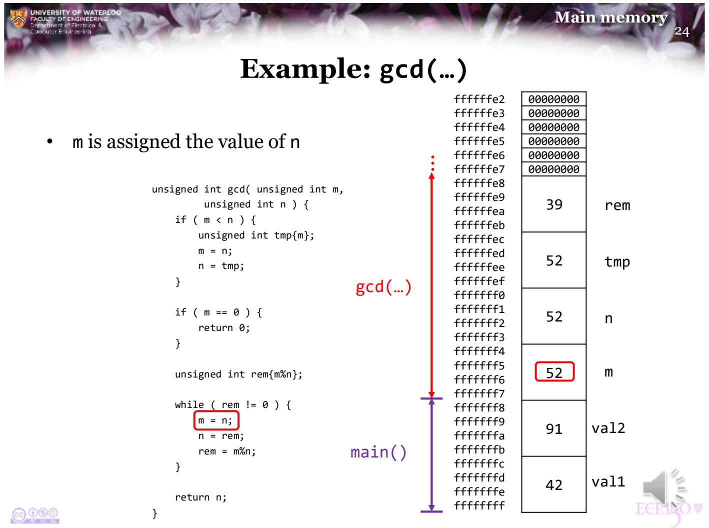
Main memory 23

Example: gcd(…)

- `m` is assigned the value of `n`

```
unsigned int gcd( unsigned int m,
         unsigned int n ) {
    if ( m < n ) {
        unsigned int tmp{m};
        m = n;
        n = tmp;
    }

    if ( m == 0 ) {
        return 0;
    }

    unsigned int rem{m%n};

    while ( rem != 0 ) {
        m = n;
        n = rem;
        rem = m%n;
    }

    return n;
}
```

ffffffe2	00000000	
ffffffe3	00000000	
ffffffe4	00000000	
ffffffe5	00000000	
ffffffe6	00000000	
ffffffe7	00000000	
ffffffe8	39	rem
ffffffe9		
ffffffea		
ffffffeb		
ffffffec	52	tmp
ffffffed		
ffffffee		
ffffffef		
fffffff0	52	n
fffffff1		
fffffff2		
fffffff3		
fffffff4	91	m
fffffff5		
fffffff6		
fffffff7		
fffffff8	91	val2
fffffff9		
fffffffa		
fffffffb		
fffffffc	42	val1
fffffffd		
fffffffe		
ffffffff		



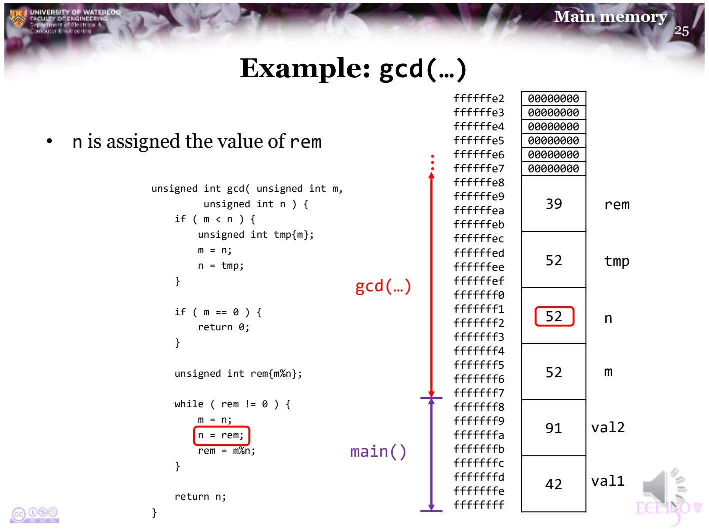
Main memory 24

Example: gcd(…)

- `m` is assigned the value of `n`

```
unsigned int gcd( unsigned int m,
         unsigned int n ) {
    if ( m < n ) {
        unsigned int tmp{m};
        m = n;
        n = tmp;
    }

    if ( m == 0 ) {
        return 0;
    }

    unsigned int rem{m%n};

    while ( rem != 0 ) {
        m = n;
        n = rem;
        rem = m%n;
    }

    return n;
}
```

ffffffe2	00000000	
ffffffe3	00000000	
ffffffe4	00000000	
ffffffe5	00000000	
ffffffe6	00000000	
ffffffe7	00000000	
ffffffe8	39	rem
ffffffe9		
ffffffea		
ffffffeb		
ffffffec	52	tmp
ffffffed		
ffffffee		
ffffffef		
fffffff0	52	n
fffffff1		
fffffff2		
fffffff3		
fffffff4	52	m
fffffff5		
fffffff6		
fffffff7		
fffffff8	91	val2
fffffff9		
fffffffa		
fffffffb		
fffffffc	42	val1
fffffffd		
fffffffe		
ffffffff		

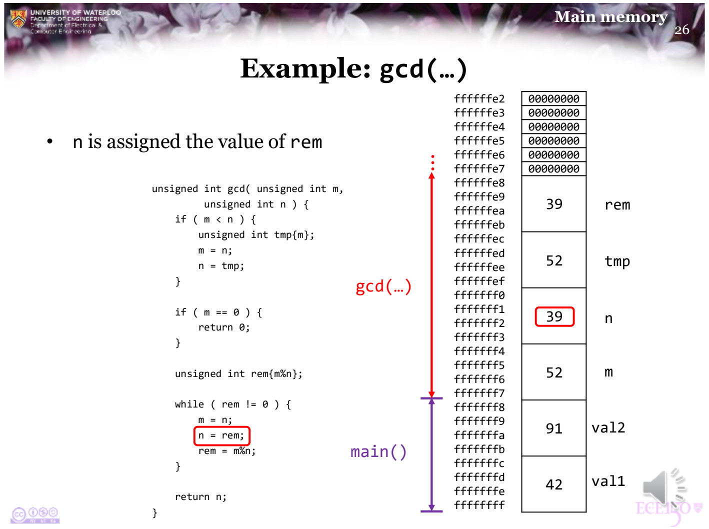


Example: gcd(…)

- n is assigned the value of rem

```
unsigned int gcd( unsigned int m,
                  unsigned int n ) {
    if ( m < n ) {
        unsigned int tmp{m};
        m = n;
        n = tmp;
    }

    if ( m == 0 ) {
        return 0;
    }

    unsigned int rem{m%n};

    while ( rem != 0 ) {
        m = n;
        n = rem;
        rem = m%n;
    }

    return n;
}
```

```
ffffffe2  00000000
ffffffe3  00000000
ffffffe4  00000000
ffffffe5  00000000
ffffffe6  00000000
ffffffe7  00000000
ffffffe8
ffffffe9     39      rem
ffffffea
ffffffeb
ffffffec
ffffffed     52      tmp
ffffffee
ffffffef
fffffff0
fffffff1     52      n
fffffff2
fffffff3
fffffff4
fffffff5     52      m
fffffff6
fffffff7
fffffff8
fffffff9     91     val2
fffffffa
fffffffb
fffffffc
fffffffd     42     val1
fffffffe
ffffffff
```

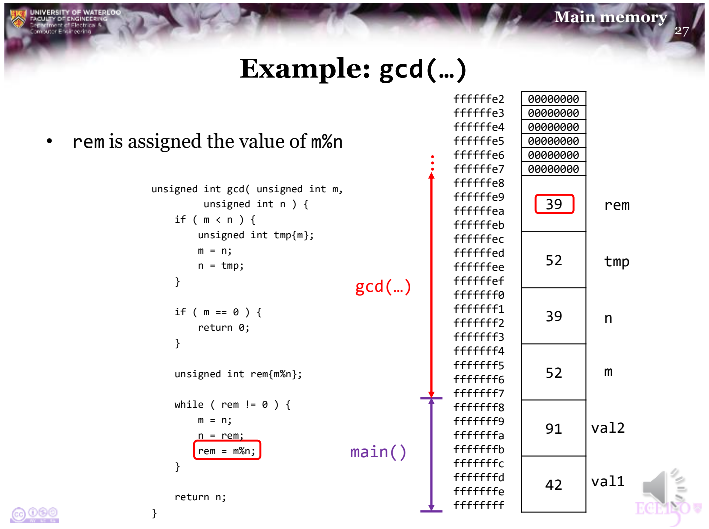


Example: gcd(…)

- n is assigned the value of rem

```
unsigned int gcd( unsigned int m,
                  unsigned int n ) {
    if ( m < n ) {
        unsigned int tmp{m};
        m = n;
        n = tmp;
    }

    if ( m == 0 ) {
        return 0;
    }

    unsigned int rem{m%n};

    while ( rem != 0 ) {
        m = n;
        n = rem;
        rem = m%n;
    }

    return n;
}
```

```
ffffffe2  00000000
ffffffe3  00000000
ffffffe4  00000000
ffffffe5  00000000
ffffffe6  00000000
ffffffe7  00000000
ffffffe8
ffffffe9     39      rem
ffffffea
ffffffeb
ffffffec
ffffffed     52      tmp
ffffffee
ffffffef
fffffff0
fffffff1     39      n
fffffff2
fffffff3
fffffff4
fffffff5     52      m
fffffff6
fffffff7
fffffff8
fffffff9     91     val2
fffffffa
fffffffb
fffffffc
fffffffd     42     val1
fffffffe
ffffffff
```

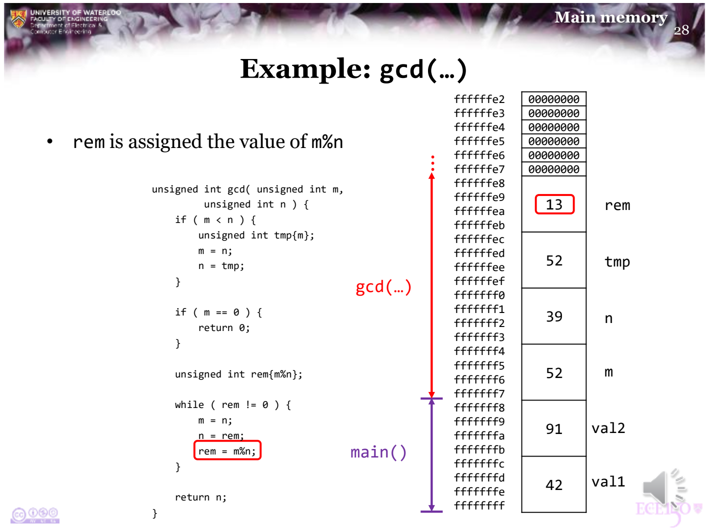


Example: gcd(…)

- rem is assigned the value of m%n

```
unsigned int gcd( unsigned int m,
                  unsigned int n ) {
    if ( m < n ) {
        unsigned int tmp{m};
        m = n;
        n = tmp;
    }

    if ( m == 0 ) {
        return 0;
    }

    unsigned int rem{m%n};

    while ( rem != 0 ) {
        m = n;
        n = rem;
        rem = m%n;
    }

    return n;
}
```

```
ffffffe2  00000000
ffffffe3  00000000
ffffffe4  00000000
ffffffe5  00000000
ffffffe6  00000000
ffffffe7  00000000
ffffffe8
ffffffe9     39      rem
ffffffea
ffffffeb
ffffffec
ffffffed     52      tmp
ffffffee
ffffffef
fffffff0
fffffff1     39      n
fffffff2
fffffff3
fffffff4
fffffff5     52      m
fffffff6
fffffff7
fffffff8
fffffff9     91     val2
fffffffa
fffffffb
fffffffc
fffffffd     42     val1
fffffffe
ffffffff
```

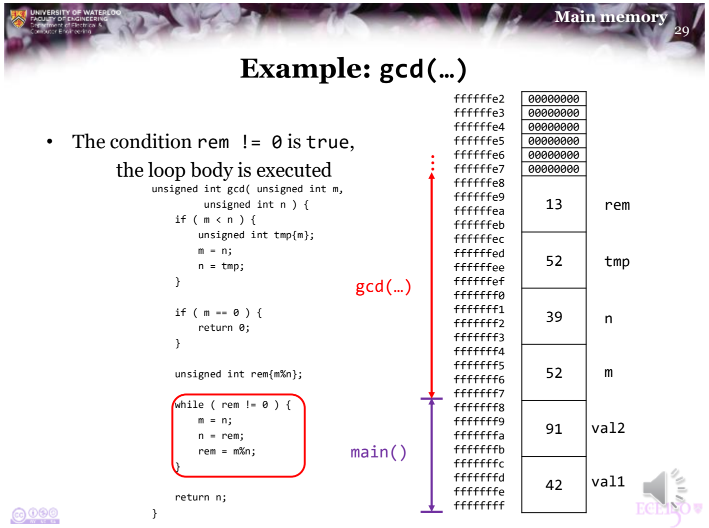


Example: gcd(…)

- rem is assigned the value of m%n

```
unsigned int gcd( unsigned int m,
                  unsigned int n ) {
    if ( m < n ) {
        unsigned int tmp{m};
        m = n;
        n = tmp;
    }

    if ( m == 0 ) {
        return 0;
    }

    unsigned int rem{m%n};

    while ( rem != 0 ) {
        m = n;
        n = rem;
        rem = m%n;
    }

    return n;
}
```

```
ffffffe2  00000000
ffffffe3  00000000
ffffffe4  00000000
ffffffe5  00000000
ffffffe6  00000000
ffffffe7  00000000
ffffffe8
ffffffe9     13      rem
ffffffea
ffffffeb
ffffffec
ffffffed     52      tmp
ffffffee
ffffffef
fffffff0
fffffff1     39      n
fffffff2
fffffff3
fffffff4
fffffff5     52      m
fffffff6
fffffff7
fffffff8
fffffff9     91     val2
fffffffa
fffffffb
fffffffc
fffffffd     42     val1
fffffffe
ffffffff
```

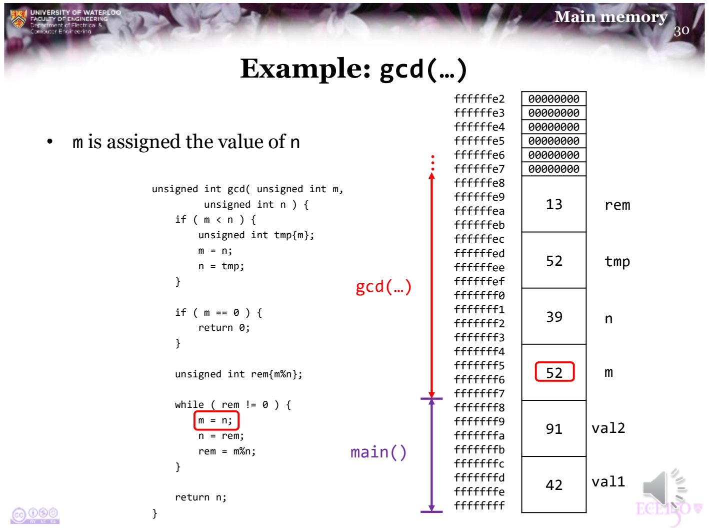


Example: gcd(…)

- The condition `rem != 0` is true, the loop body is executed

```
unsigned int gcd( unsigned int m,
                  unsigned int n ) {
    if ( m < n ) {
        unsigned int tmp{m};
        m = n;
        n = tmp;
    }

    if ( m == 0 ) {
        return 0;
    }

    unsigned int rem{m%n};

    while ( rem != 0 ) {
        m = n;
        n = rem;
        rem = m%n;
    }

    return n;
}
```

ffffffe2	00000000	
ffffffe3	00000000	
ffffffe4	00000000	
ffffffe5	00000000	
ffffffe6	00000000	
ffffffe7	00000000	
ffffffe8	13	rem
ffffffe9		
ffffffea		
ffffffeb		
ffffffec	52	tmp
ffffffed		
ffffffee		
ffffffef		
fffffff0	39	n
fffffff1		
fffffff2		
fffffff3		
fffffff4	52	m
fffffff5		
fffffff6		
fffffff7		
fffffff8	91	val2
fffffff9		
fffffffa		
fffffffb		
fffffffc	42	val1
fffffffd		
fffffffe		
ffffffff		

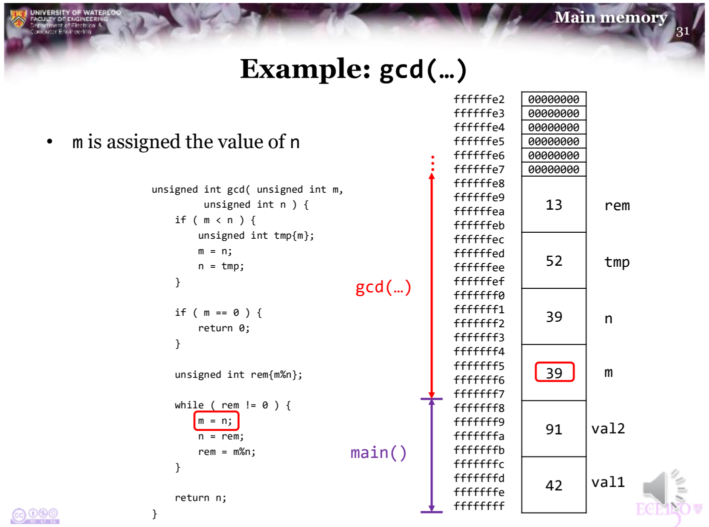


Example: gcd(…)

- `m` is assigned the value of `n`

```
unsigned int gcd( unsigned int m,
                  unsigned int n ) {
    if ( m < n ) {
        unsigned int tmp{m};
        m = n;
        n = tmp;
    }

    if ( m == 0 ) {
        return 0;
    }

    unsigned int rem{m%n};

    while ( rem != 0 ) {
        m = n;
        n = rem;
        rem = m%n;
    }

    return n;
}
```

ffffffe2	00000000	
ffffffe3	00000000	
ffffffe4	00000000	
ffffffe5	00000000	
ffffffe6	00000000	
ffffffe7	00000000	
ffffffe8	13	rem
ffffffe9		
ffffffea		
ffffffeb		
ffffffec	52	tmp
ffffffed		
ffffffee		
ffffffef		
fffffff0	39	n
fffffff1		
fffffff2		
fffffff3		
fffffff4	52	m
fffffff5		
fffffff6		
fffffff7		
fffffff8	91	val2
fffffff9		
fffffffa		
fffffffb		
fffffffc	42	val1
fffffffd		
fffffffe		
ffffffff		

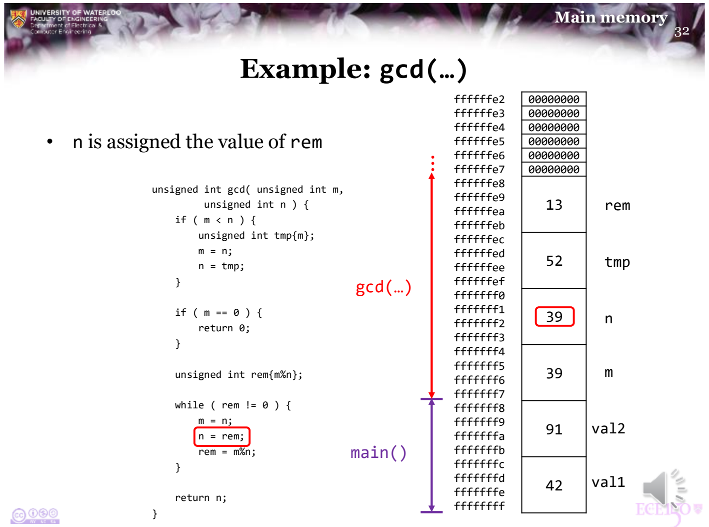


Example: gcd(…)

- `m` is assigned the value of `n`

```
unsigned int gcd( unsigned int m,
                  unsigned int n ) {
    if ( m < n ) {
        unsigned int tmp{m};
        m = n;
        n = tmp;
    }

    if ( m == 0 ) {
        return 0;
    }

    unsigned int rem{m%n};

    while ( rem != 0 ) {
        m = n;
        n = rem;
        rem = m%n;
    }

    return n;
}
```

ffffffe2	00000000	
ffffffe3	00000000	
ffffffe4	00000000	
ffffffe5	00000000	
ffffffe6	00000000	
ffffffe7	00000000	
ffffffe8	13	rem
ffffffe9		
ffffffea		
ffffffeb		
ffffffec	52	tmp
ffffffed		
ffffffee		
ffffffef		
fffffff0	39	n
fffffff1		
fffffff2		
fffffff3		
fffffff4	39	m
fffffff5		
fffffff6		
fffffff7		
fffffff8	91	val2
fffffff9		
fffffffa		
fffffffb		
fffffffc	42	val1
fffffffd		
fffffffe		
ffffffff		

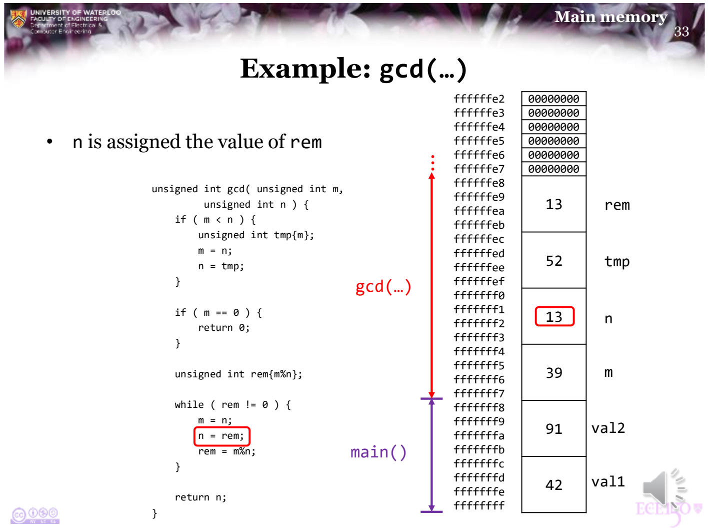


Example: gcd(…)

- `n` is assigned the value of `rem`

```
unsigned int gcd( unsigned int m,
                  unsigned int n ) {
    if ( m < n ) {
        unsigned int tmp{m};
        m = n;
        n = tmp;
    }

    if ( m == 0 ) {
        return 0;
    }

    unsigned int rem{m%n};

    while ( rem != 0 ) {
        m = n;
        n = rem;
        rem = m%n;
    }

    return n;
}
```

ffffffe2	00000000	
ffffffe3	00000000	
ffffffe4	00000000	
ffffffe5	00000000	
ffffffe6	00000000	
ffffffe7	00000000	
ffffffe8	13	rem
ffffffe9		
ffffffea		
ffffffeb		
ffffffec	52	tmp
ffffffed		
ffffffee		
ffffffef		
fffffff0	39	n
fffffff1		
fffffff2		
fffffff3		
fffffff4	39	m
fffffff5		
fffffff6		
fffffff7		
fffffff8	91	val2
fffffff9		
fffffffa		
fffffffb		
fffffffc	42	val1
fffffffd		
fffffffe		
ffffffff		

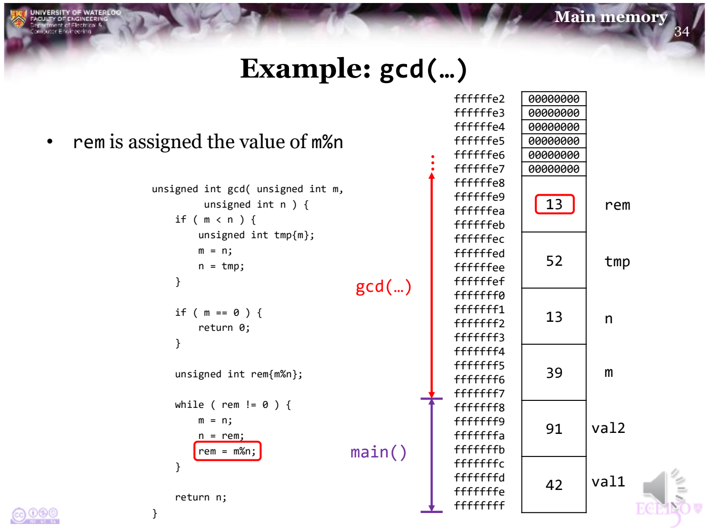


Example: gcd(…)

- n is assigned the value of rem

```
unsigned int gcd( unsigned int m,
         unsigned int n ) {
    if ( m < n ) {
        unsigned int tmp{m};
        m = n;
        n = tmp;
    }

    if ( m == 0 ) {
        return 0;
    }

    unsigned int rem{m%n};

    while ( rem != 0 ) {
        m = n;
        n = rem;
        rem = m%n;
    }

    return n;
}
```

ffffffe2	00000000	
ffffffe3	00000000	
ffffffe4	00000000	
ffffffe5	00000000	
ffffffe6	00000000	
ffffffe7	00000000	
ffffffe8	13	rem
ffffffe9		
ffffffea		
ffffffeb		
ffffffec	52	tmp
ffffffed		
ffffffee		
ffffffef		
fffffff0	13	n
fffffff1		
fffffff2		
fffffff3		
fffffff4	39	m
fffffff5		
fffffff6		
fffffff7		
fffffff8	91	val2
fffffff9		
fffffffa		
fffffffb		
fffffffc	42	val1
fffffffd		
fffffffe		
ffffffff		

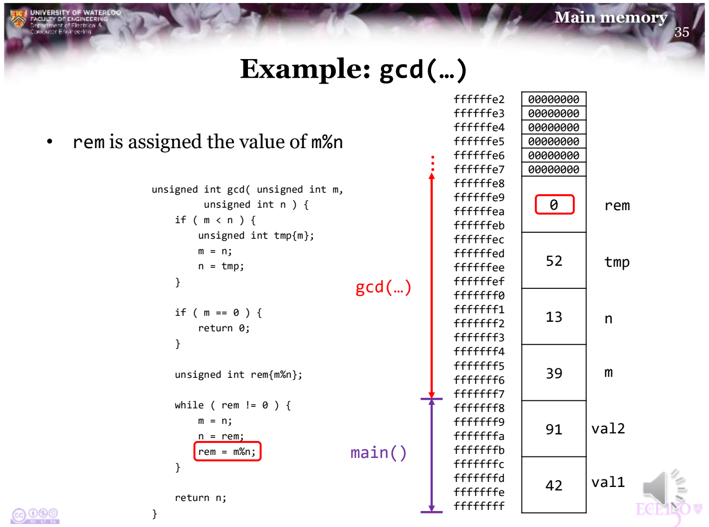


Example: gcd(…)

- rem is assigned the value of m%n

```
unsigned int gcd( unsigned int m,
         unsigned int n ) {
    if ( m < n ) {
        unsigned int tmp{m};
        m = n;
        n = tmp;
    }

    if ( m == 0 ) {
        return 0;
    }

    unsigned int rem{m%n};

    while ( rem != 0 ) {
        m = n;
        n = rem;
        rem = m%n;
    }

    return n;
}
```

ffffffe2	00000000	
ffffffe3	00000000	
ffffffe4	00000000	
ffffffe5	00000000	
ffffffe6	00000000	
ffffffe7	00000000	
ffffffe8	13	rem
ffffffe9		
ffffffea		
ffffffeb		
ffffffec	52	tmp
ffffffed		
ffffffee		
ffffffef		
fffffff0	13	n
fffffff1		
fffffff2		
fffffff3		
fffffff4	39	m
fffffff5		
fffffff6		
fffffff7		
fffffff8	91	val2
fffffff9		
fffffffa		
fffffffb		
fffffffc	42	val1
fffffffd		
fffffffe		
ffffffff		

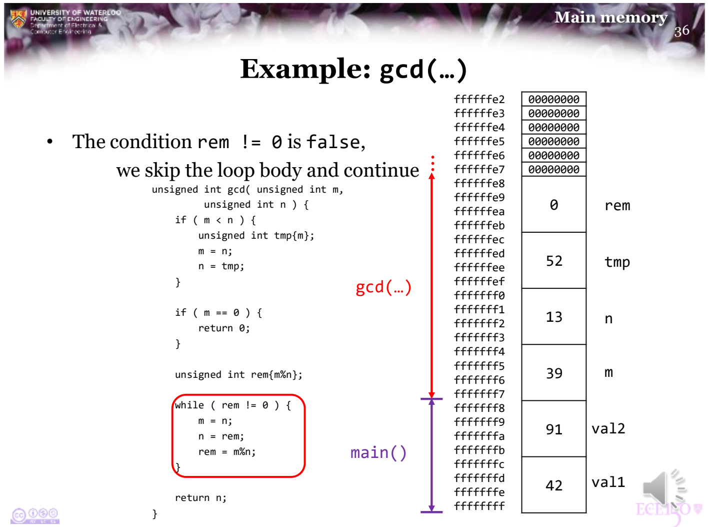


Example: gcd(…)

- rem is assigned the value of m%n

```
unsigned int gcd( unsigned int m,
         unsigned int n ) {
    if ( m < n ) {
        unsigned int tmp{m};
        m = n;
        n = tmp;
    }

    if ( m == 0 ) {
        return 0;
    }

    unsigned int rem{m%n};

    while ( rem != 0 ) {
        m = n;
        n = rem;
        rem = m%n;
    }

    return n;
}
```

ffffffe2	00000000	
ffffffe3	00000000	
ffffffe4	00000000	
ffffffe5	00000000	
ffffffe6	00000000	
ffffffe7	00000000	
ffffffe8	0	rem
ffffffe9		
ffffffea		
ffffffeb		
ffffffec	52	tmp
ffffffed		
ffffffee		
ffffffef		
fffffff0	13	n
fffffff1		
fffffff2		
fffffff3		
fffffff4	39	m
fffffff5		
fffffff6		
fffffff7		
fffffff8	91	val2
fffffff9		
fffffffa		
fffffffb		
fffffffc	42	val1
fffffffd		
fffffffe		
ffffffff		

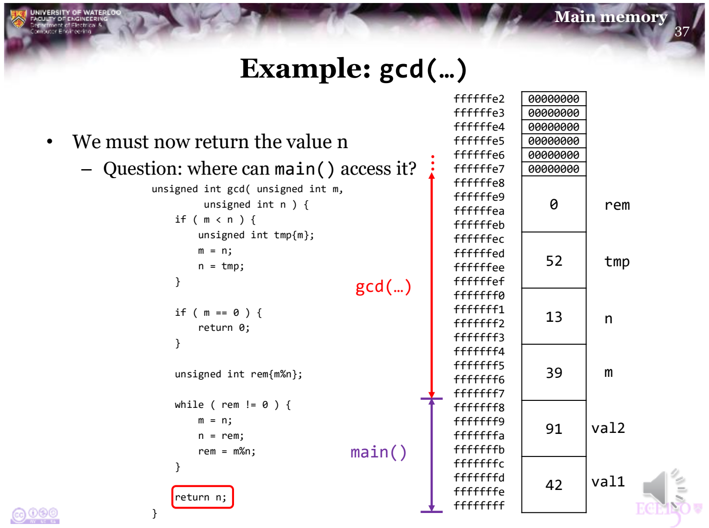


Example: gcd(…)

- The condition rem != 0 is false,
  we skip the loop body and continue

```
unsigned int gcd( unsigned int m,
         unsigned int n ) {
    if ( m < n ) {
        unsigned int tmp{m};
        m = n;
        n = tmp;
    }

    if ( m == 0 ) {
        return 0;
    }

    unsigned int rem{m%n};

    while ( rem != 0 ) {
        m = n;
        n = rem;
        rem = m%n;
    }

    return n;
}
```

ffffffe2	00000000	
ffffffe3	00000000	
ffffffe4	00000000	
ffffffe5	00000000	
ffffffe6	00000000	
ffffffe7	00000000	
ffffffe8	0	rem
ffffffe9		
ffffffea		
ffffffeb		
ffffffec	52	tmp
ffffffed		
ffffffee		
ffffffef		
fffffff0	13	n
fffffff1		
fffffff2		
fffffff3		
fffffff4	39	m
fffffff5		
fffffff6		
fffffff7		
fffffff8	91	val2
fffffff9		
fffffffa		
fffffffb		
fffffffc	42	val1
fffffffd		
fffffffe		
ffffffff		



UNIVERSITY OF WATERLOO
FACULTY OF ENGINEERING
Department of Electrical & Computer Engineering
gcd(…)
main()
ECE150

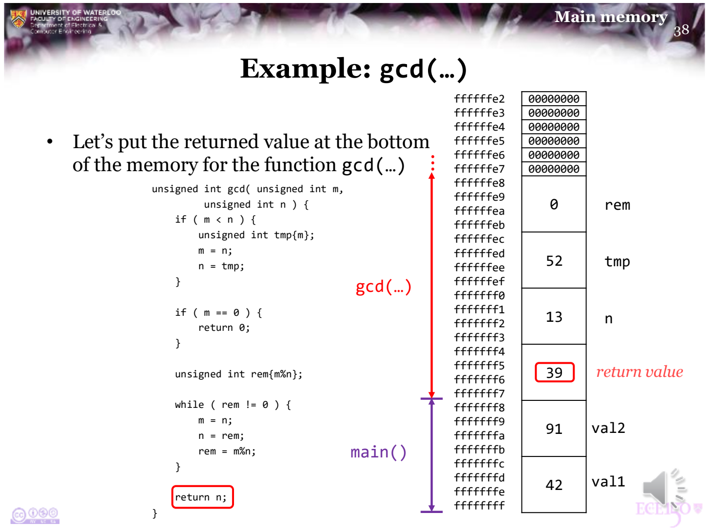
Main memory 37

Example: gcd(…)

- We must now return the value n
  - Question: where can `main()` access it?

```
unsigned int gcd( unsigned int m,
                  unsigned int n ) {
    if ( m < n ) {
        unsigned int tmp{m};
        m = n;
        n = tmp;
    }

    if ( m == 0 ) {
        return 0;
    }

    unsigned int rem{m%n};

    while ( rem != 0 ) {
        m = n;
        n = rem;
        rem = m%n;
    }

    return n;
}
```

ffffffe2	00000000	
ffffffe3	00000000	
ffffffe4	00000000	
ffffffe5	00000000	
ffffffe6	00000000	
ffffffe7	00000000	
ffffffe8	0	rem
ffffffe9		
ffffffea		
ffffffeb		
ffffffec	52	tmp
ffffffed		
ffffffee		
ffffffef		
fffffff0	13	n
fffffff1		
fffffff2		
fffffff3		
fffffff4	39	m
fffffff5		
fffffff6		
fffffff7		
fffffff8	91	val2
fffffff9		
fffffffa		
fffffffb		
fffffffc	42	val1
fffffffd		
fffffffe		
ffffffff		



UNIVERSITY OF WATERLOO
FACULTY OF ENGINEERING
Department of Electrical & Computer Engineering
gcd(…)
main()
ECE150

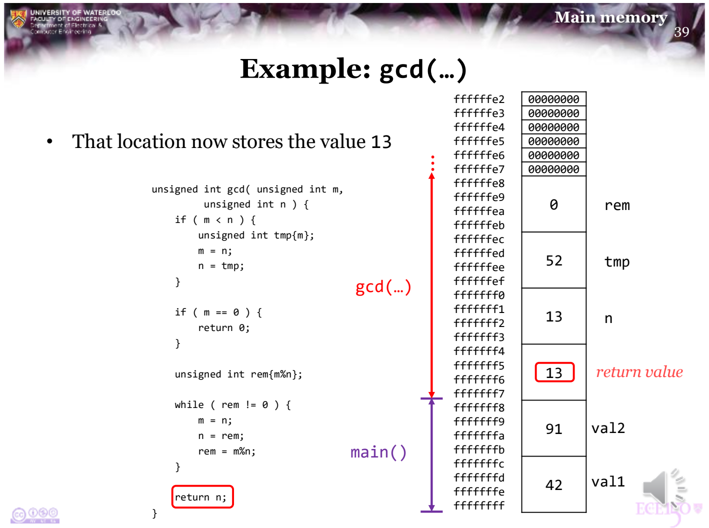
Main memory 38

Example: gcd(…)

- Let's put the returned value at the bottom of the memory for the function `gcd(…)`

```
unsigned int gcd( unsigned int m,
                  unsigned int n ) {
    if ( m < n ) {
        unsigned int tmp{m};
        m = n;
        n = tmp;
    }

    if ( m == 0 ) {
        return 0;
    }

    unsigned int rem{m%n};

    while ( rem != 0 ) {
        m = n;
        n = rem;
        rem = m%n;
    }

    return n;
}
```

ffffffe2	00000000	
ffffffe3	00000000	
ffffffe4	00000000	
ffffffe5	00000000	
ffffffe6	00000000	
ffffffe7	00000000	
ffffffe8	0	rem
ffffffe9		
ffffffea		
ffffffeb		
ffffffec	52	tmp
ffffffed		
ffffffee		
ffffffef		
fffffff0	13	n
fffffff1		
fffffff2		
fffffff3		
fffffff4	39	*return value*
fffffff5		
fffffff6		
fffffff7		
fffffff8	91	val2
fffffff9		
fffffffa		
fffffffb		
fffffffc	42	val1
fffffffd		
fffffffe		
ffffffff		



UNIVERSITY OF WATERLOO
FACULTY OF ENGINEERING
Department of Electrical & Computer Engineering
gcd(…)
main()
ECE150

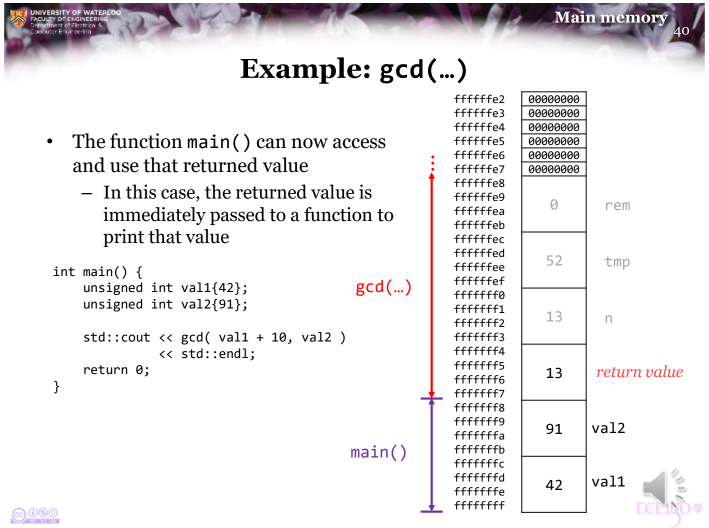
Main memory 39

Example: gcd(…)

- That location now stores the value 13

```
unsigned int gcd( unsigned int m,
                  unsigned int n ) {
    if ( m < n ) {
        unsigned int tmp{m};
        m = n;
        n = tmp;
    }

    if ( m == 0 ) {
        return 0;
    }

    unsigned int rem{m%n};

    while ( rem != 0 ) {
        m = n;
        n = rem;
        rem = m%n;
    }

    return n;
}
```

ffffffe2	00000000	
ffffffe3	00000000	
ffffffe4	00000000	
ffffffe5	00000000	
ffffffe6	00000000	
ffffffe7	00000000	
ffffffe8	0	rem
ffffffe9		
ffffffea		
ffffffeb		
ffffffec	52	tmp
ffffffed		
ffffffee		
ffffffef		
fffffff0	13	n
fffffff1		
fffffff2		
fffffff3		
fffffff4	13	*return value*
fffffff5		
fffffff6		
fffffff7		
fffffff8	91	val2
fffffff9		
fffffffa		
fffffffb		
fffffffc	42	val1
fffffffd		
fffffffe		
ffffffff		



UNIVERSITY OF WATERLOO
FACULTY OF ENGINEERING
Department of Electrical & Computer Engineering
gcd(…)
main()
ECE150

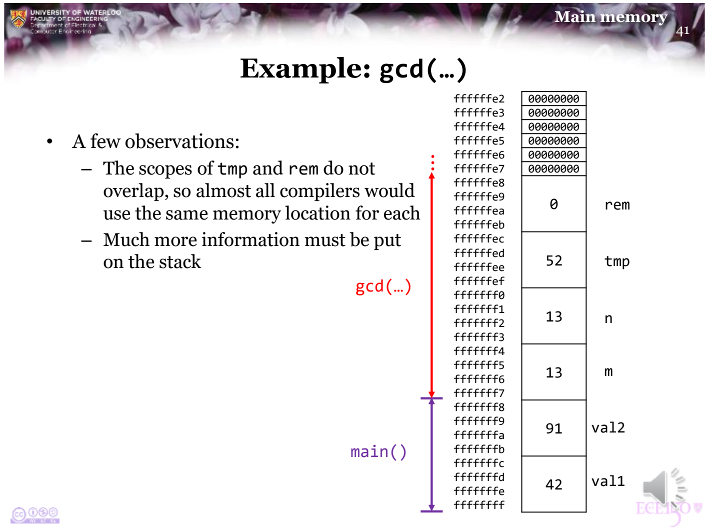
Main memory 40

Example: gcd(…)

- The function `main()` can now access and use that returned value
  - In this case, the returned value is immediately passed to a function to print that value

```
int main() {
    unsigned int val1{42};
    unsigned int val2{91};

    std::cout << gcd( val1 + 10, val2 )
              << std::endl;
    return 0;
}
```

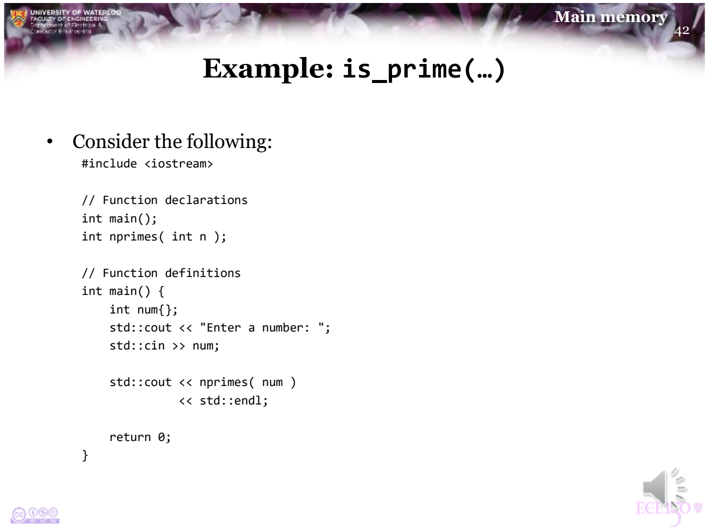
ffffffe2	00000000	
ffffffe3	00000000	
ffffffe4	00000000	
ffffffe5	00000000	
ffffffe6	00000000	
ffffffe7	00000000	
ffffffe8	0	rem
ffffffe9		
ffffffea		
ffffffeb		
ffffffec	52	tmp
ffffffed		
ffffffee		
ffffffef		
fffffff0	13	n
fffffff1		
fffffff2		
fffffff3		
fffffff4	13	*return value*
fffffff5		
fffffff6		
fffffff7		
fffffff8	91	val2
fffffff9		
fffffffa		
fffffffb		
fffffffc	42	val1
fffffffd		
fffffffe		
ffffffff		



Example: gcd(…)

- A few observations:
  - The scopes of `tmp` and `rem` do not overlap, so almost all compilers would use the same memory location for each
  - Much more information must be put on the stack

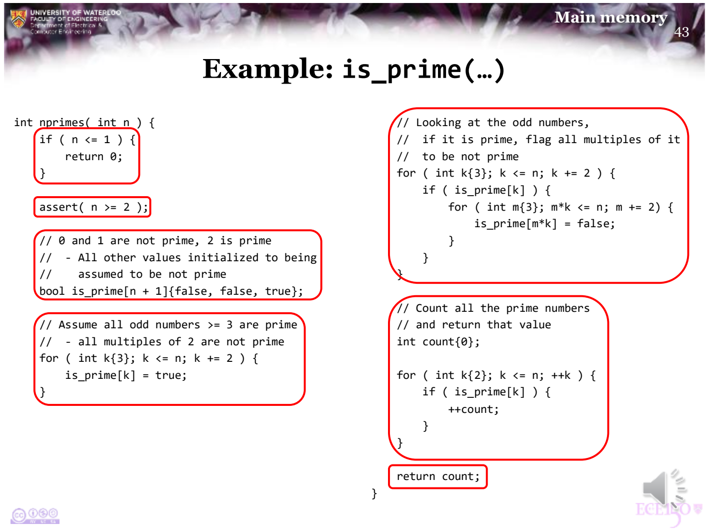
	ffffffe2	00000000	
	ffffffe3	00000000	
	ffffffe4	00000000	
	ffffffe5	00000000	
	ffffffe6	00000000	
	ffffffe7	00000000	
gcd(…)	ffffffe8	0	rem
	ffffffe9		
	ffffffea		
	ffffffeb		
	ffffffec	52	tmp
	ffffffed		
	ffffffee		
	ffffffef		
	fffffff0	13	n
	fffffff1		
	fffffff2		
	fffffff3		
	fffffff4	13	m
	fffffff5		
	fffffff6		
	fffffff7		
main()	fffffff8	91	val2
	fffffff9		
	fffffffa		
	fffffffb		
	fffffffc	42	val1
	fffffffd		
	fffffffe		
	ffffffff		



Example: is_prime(…)

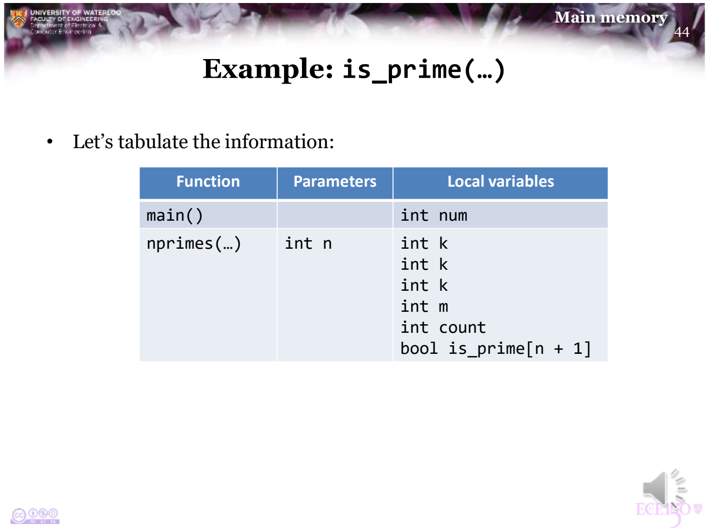
- Consider the following:

```
#include <iostream>

// Function declarations
int main();
int nprimes( int n );

// Function definitions
int main() {
    int num{};
    std::cout << "Enter a number: ";
    std::cin >> num;

    std::cout << nprimes( num )
              << std::endl;

    return 0;
}
```



Example: is_prime(…)

```
int nprimes( int n ) {
    if ( n <= 1 ) {
        return 0;
    }

    assert( n >= 2 );

    // 0 and 1 are not prime, 2 is prime
    //  - All other values initialized to being
    //    assumed to be not prime
    bool is_prime[n + 1]{false, false, true};

    // Assume all odd numbers >= 3 are prime
    //  - all multiples of 2 are not prime
    for ( int k{3}; k <= n; k += 2 ) {
        is_prime[k] = true;
    }

    // Looking at the odd numbers,
    //  if it is prime, flag all multiples of it
    //  to be not prime
    for ( int k{3}; k <= n; k += 2 ) {
        if ( is_prime[k] ) {
            for ( int m{3}; m*k <= n; m += 2) {
                is_prime[m*k] = false;
            }
        }
    }

    // Count all the prime numbers
    // and return that value
    int count{0};

    for ( int k{2}; k <= n; ++k ) {
        if ( is_prime[k] ) {
            ++count;
        }
    }

    return count;
}
```

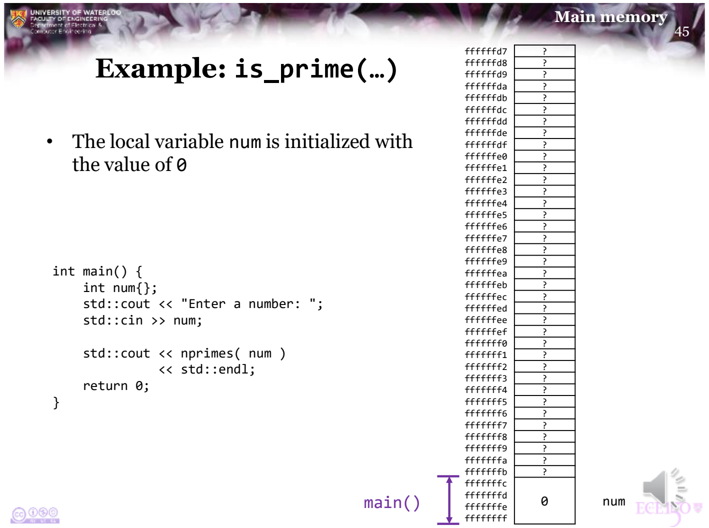


Example: is_prime(…)

- Let's tabulate the information:

Function	Parameters	Local variables
`main()`		`int num`

`nprimes(…)`	`int n`	`int k`
		`int k`
		`int k`
		`int m`
		`int count`
		`bool is_prime[n + 1]`



Main memory

45

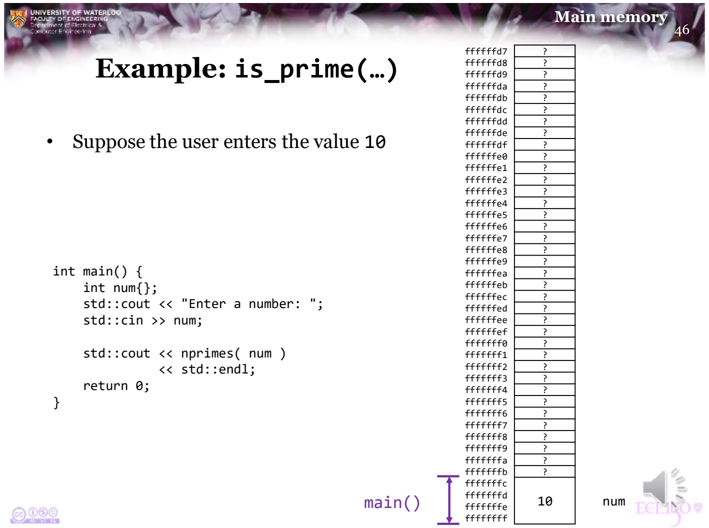
Example: is_prime(…)

- The local variable `num` is initialized with the value of `0`

```
int main() {
    int num{};
    std::cout << "Enter a number: ";
    std::cin >> num;

    std::cout << nprimes( num )
              << std::endl;
    return 0;
}
```

```
ffffffd7  ?
ffffffd8  ?
ffffffd9  ?
ffffffda  ?
ffffffdb  ?
ffffffdc  ?
ffffffdd  ?
ffffffde  ?
ffffffdf  ?
ffffffe0  ?
ffffffe1  ?
ffffffe2  ?
ffffffe3  ?
ffffffe4  ?
ffffffe5  ?
ffffffe6  ?
ffffffe7  ?
ffffffe8  ?
ffffffe9  ?
ffffffea  ?
ffffffeb  ?
ffffffec  ?
ffffffed  ?
ffffffee  ?
ffffffef  ?
fffffff0  ?
fffffff1  ?
fffffff2  ?
fffffff3  ?
fffffff4  ?
fffffff5  ?
fffffff6  ?
fffffff7  ?
fffffff8  ?
fffffff9  ?
fffffffa  ?
fffffffb  ?
fffffffc
fffffffd  0   num
fffffffe
ffffffff
```

main()



Main memory

46

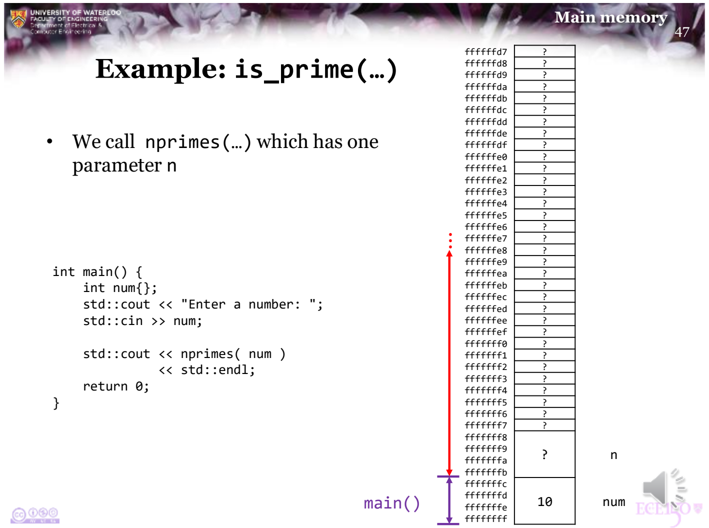
Example: is_prime(…)

- Suppose the user enters the value `10`

```
int main() {
    int num{};
    std::cout << "Enter a number: ";
    std::cin >> num;

    std::cout << nprimes( num )
              << std::endl;
    return 0;
}
```

```
ffffffd7  ?
ffffffd8  ?
ffffffd9  ?
ffffffda  ?
ffffffdb  ?
ffffffdc  ?
ffffffdd  ?
ffffffde  ?
ffffffdf  ?
ffffffe0  ?
ffffffe1  ?
ffffffe2  ?
ffffffe3  ?
ffffffe4  ?
ffffffe5  ?
ffffffe6  ?
ffffffe7  ?
ffffffe8  ?
ffffffe9  ?
ffffffea  ?
ffffffeb  ?
ffffffec  ?
ffffffed  ?
ffffffee  ?
ffffffef  ?
fffffff0  ?
fffffff1  ?
fffffff2  ?
fffffff3  ?
fffffff4  ?
fffffff5  ?
fffffff6  ?
fffffff7  ?
fffffff8  ?
fffffff9  ?
fffffffa  ?
fffffffb  ?
fffffffc
fffffffd  10  num
fffffffe
ffffffff
```

main()



Main memory

47

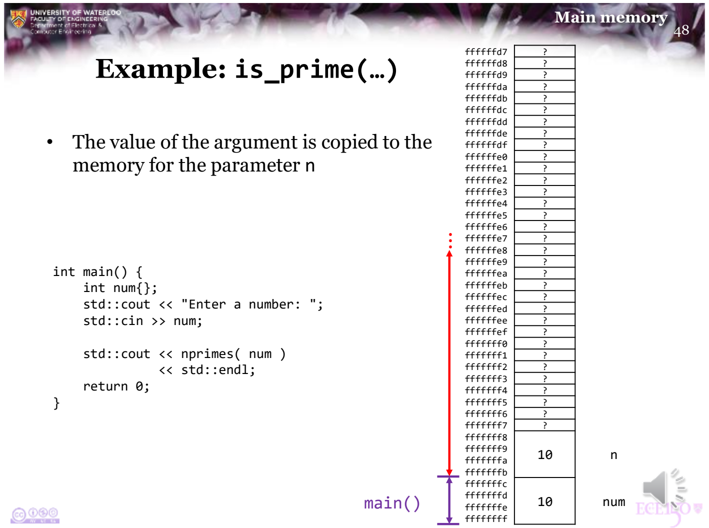
Example: is_prime(…)

- We call `nprimes(…)` which has one parameter `n`

```
int main() {
    int num{};
    std::cout << "Enter a number: ";
    std::cin >> num;

    std::cout << nprimes( num )
              << std::endl;
    return 0;
}
```

```
ffffffd7  ?
ffffffd8  ?
ffffffd9  ?
ffffffda  ?
ffffffdb  ?
ffffffdc  ?
ffffffdd  ?
ffffffde  ?
ffffffdf  ?
ffffffe0  ?
ffffffe1  ?
ffffffe2  ?
ffffffe3  ?
ffffffe4  ?
ffffffe5  ?
ffffffe6  ?
ffffffe7  ?
ffffffe8  ?
ffffffe9  ?
ffffffea  ?
ffffffeb  ?
ffffffec  ?
ffffffed  ?
ffffffee  ?
ffffffef  ?
fffffff0  ?
fffffff1  ?
fffffff2  ?
fffffff3  ?
fffffff4  ?
fffffff5  ?
fffffff6  ?
fffffff7  ?
fffffff8
fffffff9  ?   n
fffffffa
fffffffb
fffffffc
fffffffd  10  num
fffffffe
ffffffff
```

main()



Main memory

48

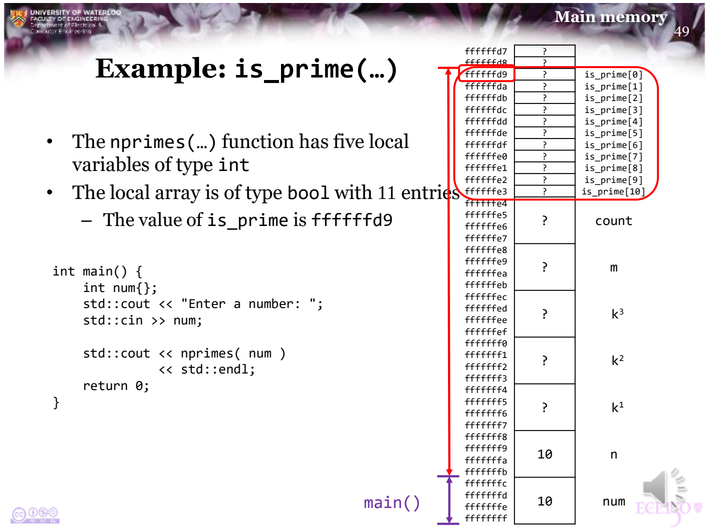
Example: is_prime(…)

- The value of the argument is copied to the memory for the parameter `n`

```
int main() {
    int num{};
    std::cout << "Enter a number: ";
    std::cin >> num;

    std::cout << nprimes( num )
              << std::endl;
    return 0;
}
```

```
ffffffd7  ?
ffffffd8  ?
ffffffd9  ?
ffffffda  ?
ffffffdb  ?
ffffffdc  ?
ffffffdd  ?
ffffffde  ?
ffffffdf  ?
ffffffe0  ?
ffffffe1  ?
ffffffe2  ?
ffffffe3  ?
ffffffe4  ?
ffffffe5  ?
ffffffe6  ?
ffffffe7  ?
ffffffe8  ?
ffffffe9  ?
ffffffea  ?
ffffffeb  ?
ffffffec  ?
ffffffed  ?
ffffffee  ?
ffffffef  ?
fffffff0  ?
fffffff1  ?
fffffff2  ?
fffffff3  ?
fffffff4  ?
fffffff5  ?
fffffff6  ?
fffffff7  ?
fffffff8
fffffff9  10  n
fffffffa
fffffffb
fffffffc
fffffffd  10  num
fffffffe
ffffffff
```

main()



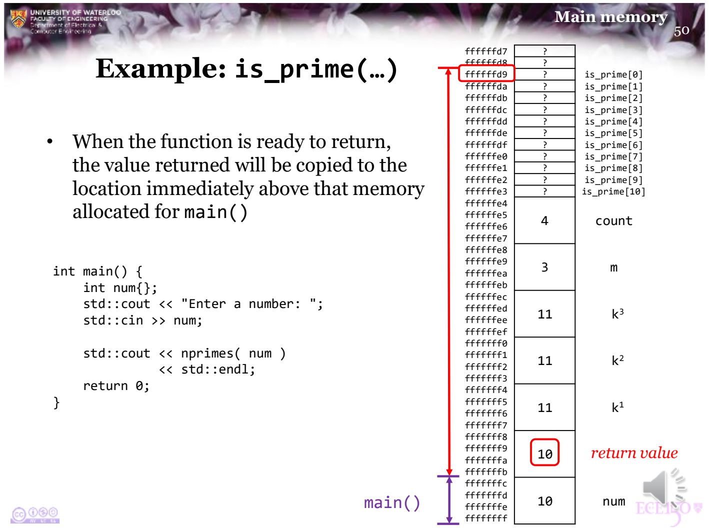
Example: is_prime(…)

- The nprimes(…) function has five local variables of type int
- The local array is of type bool with 11 entries
  - The value of is_prime is ffffffd9

```
int main() {
    int num{};
    std::cout << "Enter a number: ";
    std::cin >> num;

    std::cout << nprimes( num )
              << std::endl;
    return 0;
}
```

ffffffd7	?	
ffffffd8	?	
ffffffd9	?	is_prime[0]
ffffffda	?	is_prime[1]
ffffffdb	?	is_prime[2]
ffffffdc	?	is_prime[3]
ffffffdd	?	is_prime[4]
ffffffde	?	is_prime[5]
ffffffdf	?	is_prime[6]
ffffffe0	?	is_prime[7]
ffffffe1	?	is_prime[8]
ffffffe2	?	is_prime[9]
ffffffe3	?	is_prime[10]
ffffffe4 ffffffe5 ffffffe6 ffffffe7	?	count
ffffffe8 ffffffe9 ffffffea ffffffeb	?	m
ffffffec ffffffed ffffffee ffffffef	?	k^3
fffffff0 fffffff1 fffffff2 fffffff3	?	k^2
fffffff4 fffffff5 fffffff6 fffffff7	?	k^1
fffffff8 fffffff9 fffffffa fffffffb	10	n
fffffffc fffffffd fffffffe ffffffff	10	num

main()



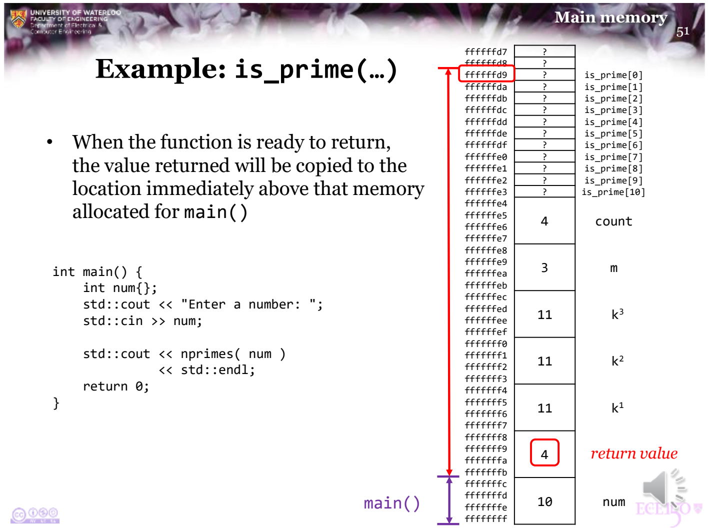
Example: is_prime(…)

- When the function is ready to return, the value returned will be copied to the location immediately above that memory allocated for main()

```
int main() {
    int num{};
    std::cout << "Enter a number: ";
    std::cin >> num;

    std::cout << nprimes( num )
              << std::endl;
    return 0;
}
```

ffffffd7	?	
ffffffd8	?	
ffffffd9	?	is_prime[0]
ffffffda	?	is_prime[1]
ffffffdb	?	is_prime[2]
ffffffdc	?	is_prime[3]
ffffffdd	?	is_prime[4]
ffffffde	?	is_prime[5]
ffffffdf	?	is_prime[6]
ffffffe0	?	is_prime[7]
ffffffe1	?	is_prime[8]
ffffffe2	?	is_prime[9]
ffffffe3	?	is_prime[10]
ffffffe4 ffffffe5 ffffffe6 ffffffe7	4	count
ffffffe8 ffffffe9 ffffffea ffffffeb	3	m
ffffffec ffffffed ffffffee ffffffef	11	k^3
fffffff0 fffffff1 fffffff2 fffffff3	11	k^2
fffffff4 fffffff5 fffffff6 fffffff7	11	k^1
fffffff8 fffffff9 fffffffa fffffffb	10	*return value*
fffffffc fffffffd fffffffe ffffffff	10	num

main()



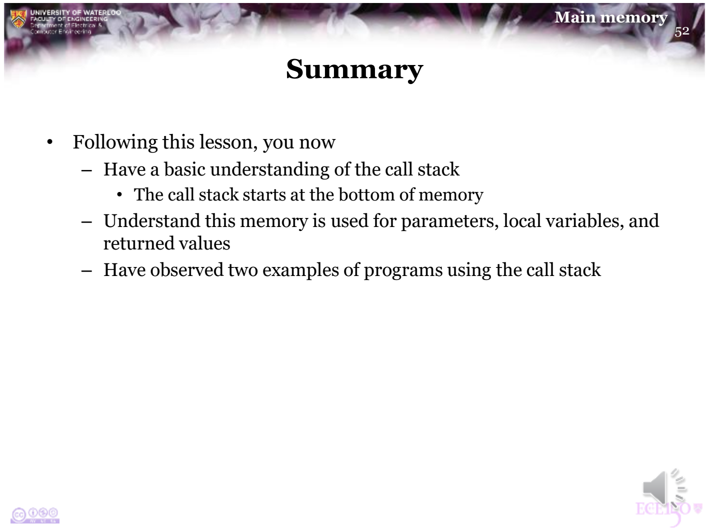
Example: is_prime(…)

- When the function is ready to return, the value returned will be copied to the location immediately above that memory allocated for main()

```
int main() {
    int num{};
    std::cout << "Enter a number: ";
    std::cin >> num;

    std::cout << nprimes( num )
              << std::endl;
    return 0;
}
```

ffffffd7	?	
ffffffd8	?	
ffffffd9	?	is_prime[0]
ffffffda	?	is_prime[1]
ffffffdb	?	is_prime[2]
ffffffdc	?	is_prime[3]
ffffffdd	?	is_prime[4]
ffffffde	?	is_prime[5]
ffffffdf	?	is_prime[6]
ffffffe0	?	is_prime[7]
ffffffe1	?	is_prime[8]
ffffffe2	?	is_prime[9]
ffffffe3	?	is_prime[10]
ffffffe4 ffffffe5 ffffffe6 ffffffe7	4	count
ffffffe8 ffffffe9 ffffffea ffffffeb	3	m
ffffffec ffffffed ffffffee ffffffef	11	k^3
fffffff0 fffffff1 fffffff2 fffffff3	11	k^2
fffffff4 fffffff5 fffffff6 fffffff7	11	k^1
fffffff8 fffffff9 fffffffa fffffffb	4	*return value*
fffffffc fffffffd fffffffe ffffffff	10	num

main()

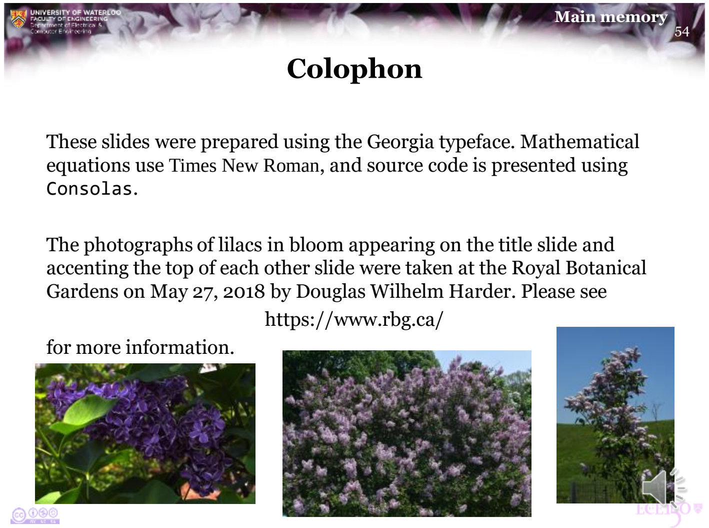


Summary

- Following this lesson, you now
  - Have a basic understanding of the call stack
    - The call stack starts at the bottom of memory
  - Understand this memory is used for parameters, local variables, and returned values
  - Have observed two examples of programs using the call stack

References

[1] Wikipedia:
https://en.wikipedia.org/wiki/Call_stack



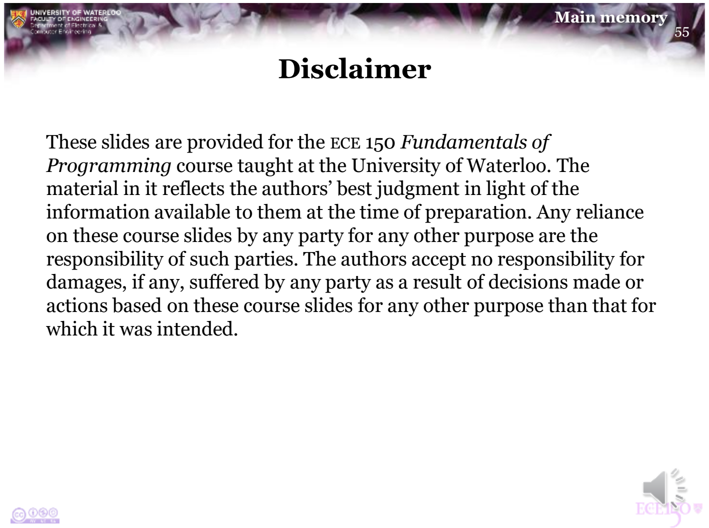
Colophon

These slides were prepared using the Georgia typeface. Mathematical equations use Times New Roman, and source code is presented using `Consolas`.

The photographs of lilacs in bloom appearing on the title slide and accenting the top of each other slide were taken at the Royal Botanical Gardens on May 27, 2018 by Douglas Wilhelm Harder. Please see

https://www.rbg.ca/

for more information.



Disclaimer

These slides are provided for the ECE 150 *Fundamentals of Programming* course taught at the University of Waterloo. The material in it reflects the authors' best judgment in light of the information available to them at the time of preparation. Any reliance on these course slides by any party for any other purpose are the responsibility of such parties. The authors accept no responsibility for damages, if any, suffered by any party as a result of decisions made or actions based on these course slides for any other purpose than that for which it was intended.