Recursive method tracing
Stack based approach
Nested recursive call example

By: Brandon Horn

mrHorn.com, Inc.

What does `mystery(45)` return?

```
public static int mystery(int value)
{
  if(value <= 10)
    return value * 3;

  return value + mystery(mystery(value / 5));
}
```

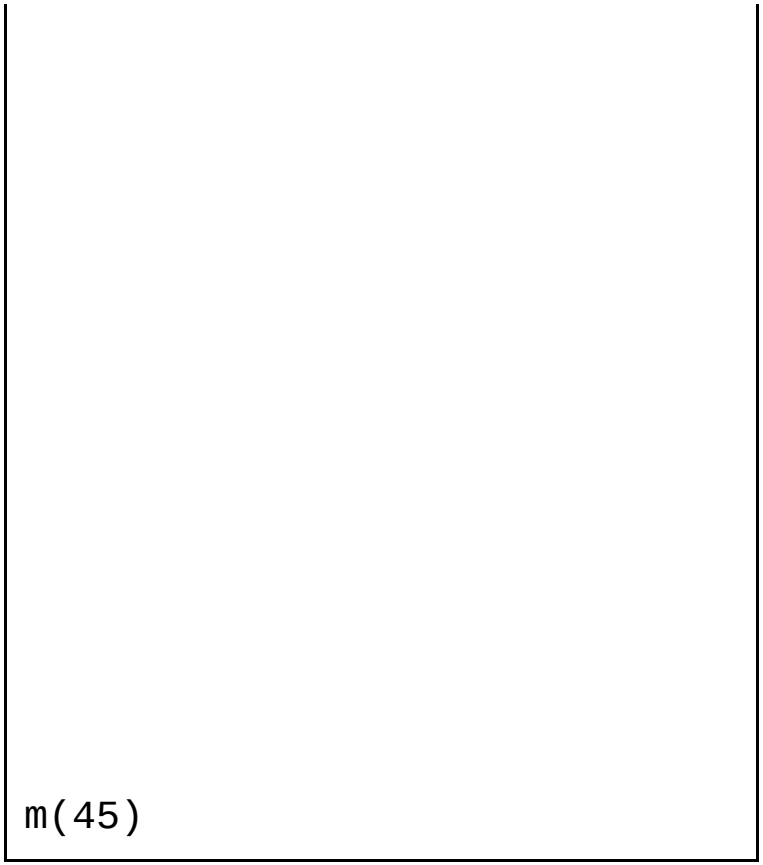
There are 2 recursive calls in `mystery`.
Label the inner recursive call 1 and the outer recursive call 2.

```
public static int mystery(int value)
{
  if(value <= 10)
    return value * 3;

  return value + mystery(mystery(value / 5));
}
```

Call 2 Call 1

Use a stack to keep track of the method calls .
Abbreviate the method name as `m`.
The initial call is `m(45)`.

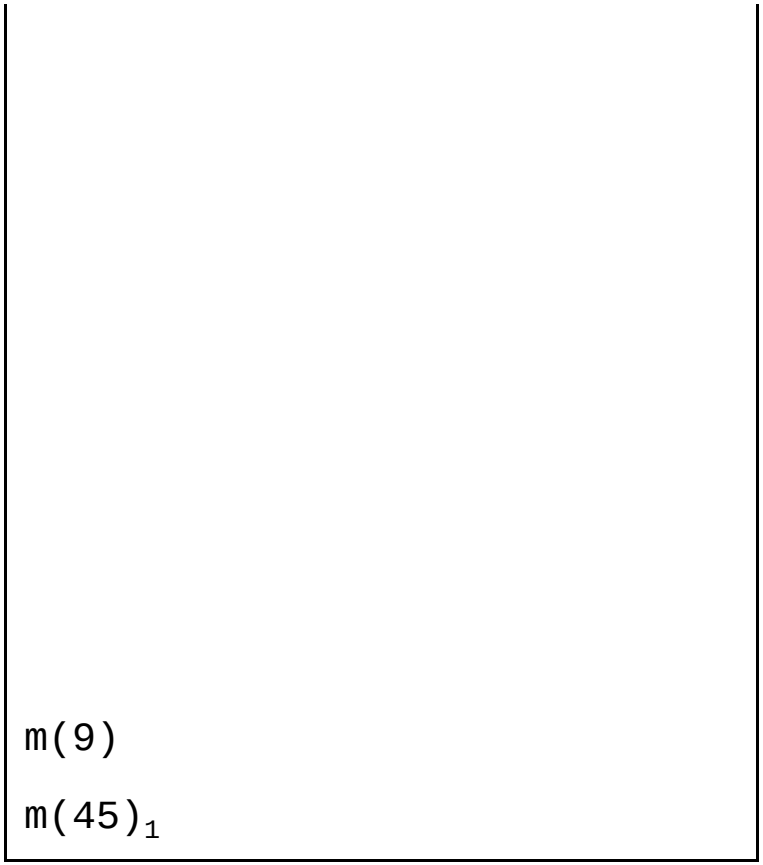



```
public static int mystery(int value)
{
  if(value <= 10)
    return value * 3;

  return value + mystery(mystery(value / 5));
}
```

Call 2 Call 1

`m(45)` stops at call 1 and calls `m(9)`.
Track where m(45) stops with a subscript next to the call.
Add the recursive call to the top of the stack.




```
public static int mystery(int value)
{
  if(value <= 10)
    return value * 3;

  return value + mystery(mystery(value / 5));
}
```

Call 2 Call 1

`m(9)` returns 27.
Cross out `m(9)` and note the return value.

~~`m(9)`~~ returns 27

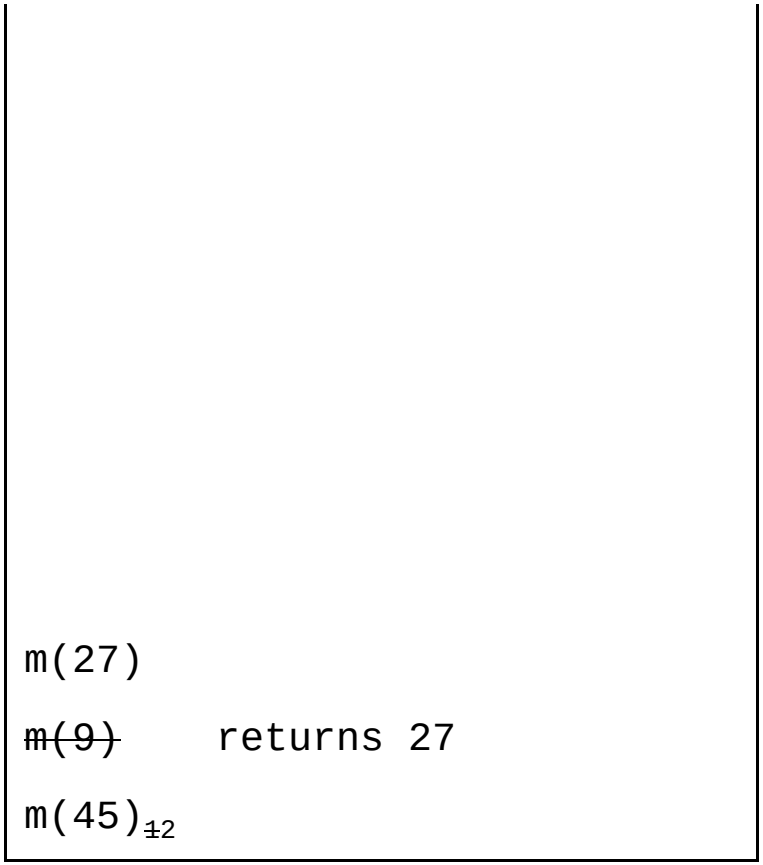
$m(45)_1$

```
public static int mystery(int value)
{
  if(value <= 10)
    return value * 3;

  return value + mystery(mystery(value / 5));
}
```

Call 2 Call 1

Control returns to `m(45)`, the topmost method on the stack that has not yet returned a value.
`m(45)` has just finished call 1 and got back 27.
`m(45)` now stops at call 2 and calls `m(27)`.
Cross out the 1 and replace it with a 2.
Add the call to `m(27)` to the top of the stack.




```
public static int mystery(int value)
{
  if(value <= 10)
    return value * 3;

  return value + mystery(mystery(value / 5));
}
```

Call 2 Call 1

m(5)

m(27)$_1$

~~m(9)~~ returns 27

m(45)$_{\text{~~1~~2}}$

```
public static int mystery(int value)
{
  if(value <= 10)
    return value * 3;

  return value + mystery(mystery(value / 5));
}
```

Call 2 Call 1

~~m(5)~~ returns 15

$m(27)_1$

~~m(9)~~ returns 27

$m(45)_{\sim\!\!\sim 2}$

```
public static int mystery(int value)
{
  if(value <= 10)
    return value * 3;

  return value + mystery(mystery(value / 5));
}
```

Call 2 Call 1

m(15)

~~m(5)~~ returns 15

m(27)$_{\text{+12}}$

~~m(9)~~ returns 27

m(45)$_{\text{+12}}$

```
public static int mystery(int value)
{
  if(value <= 10)
    return value * 3;

  return value + mystery(mystery(value / 5));
}
```

Call 2 Call 1

m(3)

m(15)$_1$

~~m(5)~~ returns 15

m(27)$_{1 2}$

~~m(9)~~ returns 27

m(45)$_{1 2}$

```
public static int mystery(int value)
{
  if(value <= 10)
    return value * 3;

  return value + mystery(mystery(value / 5));
}
```

Call 2 Call 1

~~m(3)~~ returns 9

$m(15)_1$

~~m(5)~~ returns 15

$m(27)_{12}$

~~m(9)~~ returns 27

$m(45)_{12}$

```
public static int mystery(int value)
{
  if(value <= 10)
    return value * 3;

  return value + mystery(mystery(value / 5));
}
```

Call 2 Call 1

m(9)

~~m(3)~~ returns 9

$m(15)_{\text{~~12~~}}$

~~m(5)~~ returns 15

$m(27)_{\text{~~12~~}}$

~~m(9)~~ returns 27

$m(45)_{\text{~~12~~}}$

```
public static int mystery(int value)
{
  if(value <= 10)
    return value * 3;

  return value + mystery(mystery(value / 5));
}
```

Call 2 Call 1

~~m(9)~~ returns 27

~~m(3)~~ returns 9

m(15)$_{\text{1}\cancel{2}}$

~~m(5)~~ returns 15

m(27)$_{\text{1}\cancel{2}}$

~~m(9)~~ returns 27

m(45)$_{\text{1}\cancel{2}}$

```
public static int mystery(int value)
{
  if(value <= 10)
    return value * 3;

  return value + mystery(mystery(value / 5));
}
```

Call 2 Call 1

`m(15)` just finished call 2 and got back 27.
`m(15)` returns the sum of `value` and the return value from call 2.

~~`m(9)`~~ returns 27

~~`m(3)`~~ returns 9

~~`m(15)`~~$_{\text{~~12~~}}$ returns 15 + 27 = 42

~~`m(5)`~~ returns 15

`m(27)`$_{\text{~~12~~}}$

~~`m(9)`~~ returns 27

`m(45)`$_{\text{~~12~~}}$

```
public static int mystery(int value)
{
  if(value <= 10)
    return value * 3;

  return value + mystery(mystery(value / 5));
}
```

Call 2 Call 1

~~m(9)~~ returns 27

~~m(3)~~ returns 9

~~m(15)~~$_{12}$ returns 15 + 27 = 42

~~m(5)~~ returns 15

~~m(27)~~$_{12}$ returns 27 + 42 = 69

~~m(9)~~ returns 27

m(45)$_{12}$

```
public static int mystery(int value)
{
  if(value <= 10)
    return value * 3;

  return value + mystery(mystery(value / 5));
}
```

Call 2 Call 1

~~m(9)~~ returns 27

~~m(3)~~ returns 9

~~m(15)~~$_{12}$ returns 15 + 27 = 42

~~m(5)~~ returns 15

~~m(27)~~$_{12}$ returns 27 + 42 = 69

~~m(9)~~ returns 27

~~m(45)~~$_{12}$ returns 45 + 69 = 114

```
public static int mystery(int value)
{
  if(value <= 10)
    return value * 3;

  return value + mystery(mystery(value / 5));
}
```

Call 2 Call 1