Recursive Example

_factorial_

_just before the call to factorial_

```assembly
0x08048054 <+0>:  push  $0x5
0x08048056 <+2>:  call  0x8048067 <factorial>
0x0804805b <+7>:  add   $0x4,%esp
0x0804805e <+10>: mov   %eax,%ebx
0x08048060 <+12>: mov   $0x1,%eax
0x08048065 <+17>: int   $0x80
```

Stack
```assembly
0xffffd66c
```

```assembly
0x8048067 <+0>:  push  %ebp
0x8048068 <+1>:  mov   %esp,%ebp
0x804806a <+3>:  mov   0x8(%ebp),%eax
0x804806d <+6>:  cmp   $0x1,%eax
0x8048070 <+9>:  je    0x804807f <end_factorial>
0x8048072 <+11>: dec   %eax
0x8048073 <+12>: push  %eax
0x8048074 <+13>: call  0x8048067 <factorial>
0x8048079 <+18>: mov   0x8(%ebp),%ebx
0x804807c <+21>: imul  %ebx,%eax
```
just after the call to factorial(5)

```
0x0804805b 5 0x00000000 0xffffd668 0xffffd66c 0xffffd664
```

```
Stack
```

```text
line in _start after call to factorial
old EBP
Stack
```

about to call factorial(4)

```
0x00000000 4 0x0804805b 5 0xffffd660 0xffffd66c 0xffffd664 0xffffd668
```

```
Stack
```

```text
line in factorial after call to factorial
old EBP
```

about to call factorial(3)

```
0xffffd660 4 0x00000000
```

```
Stack
```

```text
line in _start after call to factorial
old EBP
```

about to call factorial(2)

```
0xffffd660 4 0x00000000
```

```
Stack
```

```text
line in factorial after call to factorial
old EBP
```

```
Stack
```

```text
line in _start after call to factorial
old EBP
```

```
Stack
```

```text
line in factorial after call to factorial
old EBP
```

```
Stack
```

```text
line in _start after call to factorial
old EBP
```
How did we get these values? GDB.

- to get the code in assembly:
  - disas label e.g.,
    - disas _start
    - disas factorial
- to read the value stored in register %ebp
  - p $ebp
- to read a series of 20 values stored on the stack starting with the value stored in %esp:
  - x/20x $esp