

# NESTED (RECURSIVE) FUNCTION IN RISC-V ISA

Dr Sparsh Mittal

# Nested functions

- For nested call, caller needs to save on the stack:
  - Its return address
  - Any arguments and temporaries needed after the call
- Restore from the stack after the call

# Convert this to RISC-V

- C code:

```
long long int fact (long long int n)
{
    if (n < 1) return 1;
    else return n * fact(n - 1);
}
```

- Argument n is initially stored in x10
- Final result should be stored in x10
- “jr x1” is same as “jalr x0 x1 0”.

Example instruction	Instruction name	Meaning
jal x1,offset	Jump and link	$\text{Regs}[\text{x1}] \leftarrow \text{PC} + 4; \text{PC} \leftarrow \text{PC} + (\text{offset} \ll 1)$
jalr x1,x2,offset	Jump and link register	$\text{Regs}[\text{x1}] \leftarrow \text{PC} + 4; \text{PC} \leftarrow \text{Regs}[\text{x2}] + \text{offset}$

# Example

```
j main
fact:
    addi sp,sp,-16          Save return address and n on stack
    sd    x1,8(sp)
    sd    x10,0(sp)
    addi  x5,x10,-1         x5 = n - 1
    bge   x5,x0,L1          if n >= 0, go to L1
    addi  x10,x0,1          Else, set return value to 1
    addi  sp,sp,16          Pop stack, don't bother restoring values
    jr    x1                Return
L1: addi  x10,x10,-1         n = n - 1
    jal   x1,fact            call fact(n-1)
    addi  x6,x10,0           move result of fact(n - 1) to x6
    ld    x10,0(sp)          Restore caller's n
    ld    x1,8(sp)           Restore caller's return address
    addi  sp,sp,16          Pop stack
    mul   x10,x10,x6         return n * fact(n-1)
    jr    x1                return
main:
    addi  x10, x0, 5         ←
    jal   x1, fact #addr=2000
```

# Example

fact:

addi sp,sp,-16  Save return address and n on stack

sd x1,8(sp)

sd x10,0(sp)

addi x5,x10,-1 x5 = n - 1

bge x5,x0,L1 if n >= 0, go to L1

addi x10,x0,1 Else, set return value to 1

addi sp,sp,16 Pop stack, don't bother restoring values

jr x1 Return

L1: addi x10,x10,-1 n = n - 1

jal x1,fact call fact(n-1)

addi x6,x10,0 move result of fact(n - 1) to x6

ld x10,0(sp) Restore caller's n

ld x1,8(sp) Restore caller's return address

addi sp,sp,16 Pop stack

mul x10,x10,x6 return n \* fact(n-1)

jr x1 return

main:

addi x10, x0, 5

jal x1, fact #addr=2000

x10	5
x2 (sp)	1000
x1 (ra)	2004

Registers

936	
944	
952	
960	
968	
976	
984	
992	
1000	

Memory

# Example

fact:

addi sp,sp,-16

Save return address and n on stack

sd x1,8(sp)

sd x10,0(sp)

addi x5,x10,-1



x5 = n - 1

bge x5,x0,L1

if n >= 0, go to L1

addi x10,x0,1

Else, set return value to 1

addi sp,sp,16

Pop stack, don't bother restoring values

jr x1

Return

L1: addi x10,x10,-1

n = n - 1

jal x1,fact

call fact(n-1)

addi x6,x10,0

move result of fact(n - 1) to x6

ld x10,0(sp)

Restore caller's n

ld x1,8(sp)

Restore caller's return address

addi sp,sp,16

Pop stack

mul x10,x10,x6

return n \* fact(n-1)

jr x1

return

main:

addi x10, x0, 5

jal x1, fact #addr=2000

x5	
x10	5

x2 (sp)	984
x1 (ra)	2004

936	
944	
952	
960	
968	
976	
984	5
992	2004
1000	

# Example

fact:

addi sp,sp,-16

Save return address and n on stack

sd x1,8(sp)

sd x10,0(sp)

addi x5,x10,-1

$x5 = n - 1$

bge x5,x0,L1

if  $n \geq 0$ , go to L1

addi x10,x0,1

Else, set return value to 1

addi sp,sp,16

Pop stack, don't bother restoring values

jr x1

Return

L1: addi x10,x10,-1

$n = n - 1$

jal x1,fact

call fact(n-1)

addi x6,x10,0

move result of fact(n - 1) to x6

ld x10,0(sp)

Restore caller's n

ld x1,8(sp)

Restore caller's return address

addi sp,sp,16

Pop stack

mul x10,x10,x6

return  $n * \text{fact}(n-1)$

jr x1

return

main:

addi x10, x0, 5

jal x1, fact #addr=2000

x5	4
x10	5

x2 (sp)	984
x1 (ra)	2004

936	
944	
952	
960	
968	
976	
984	5
992	2004
1000	

# Example

fact:

addi sp,sp,-16                      Save return address and n on stack

sd x1,8(sp)

sd x10,0(sp)

addi x5,x10,-1                      x5 = n - 1

bge x5,x0,L1                      if n >= 0, go to L1

addi x10,x0,1                      Else, set return value to 1

addi sp,sp,16                      Pop stack, don't bother restoring values

jr x1                      Return

L1: addi x10,x10,-1                      ← n = n - 1

jal x1,fact                      call fact(n-1)

addi x6,x10,0                      move result of fact(n - 1) to x6

ld x10,0(sp)                      Restore caller's n

ld x1,8(sp)                      Restore caller's return address

addi sp,sp,16                      Pop stack

mul x10,x10,x6                      return n \* fact(n-1)

jr x1                      return

main:

addi x10, x0, 5

jal x1, fact #addr=2000

x5	4
x10	5

x2 (sp)	984
x1 (ra)	2004

936	
944	
952	
960	
968	
976	
984	5
992	2004
1000	



# Example

fact:

addi sp,sp,-16

Save return address and n on stack

sd x1,8(sp)

sd x10,0(sp)

addi x5,x10,-1

$x5 = n - 1$

bge x5,x0,L1

if  $n \geq 0$ , go to L1

addi x10,x0,1

Else, set return value to 1

addi sp,sp,16

Pop stack, don't bother restoring values

jr x1

Return

L1: addi x10,x10,-1

$n = n - 1$

jal x1,fact

call fact( $n-1$ )

addi x6,x10,0

move result of fact( $n - 1$ ) to x6

ld x10,0(sp)

Restore caller's n

ld x1,8(sp)

Restore caller's return address

addi sp,sp,16

Pop stack

mul x10,x10,x6

return  $n * \text{fact}(n-1)$

jr x1

return

main:

addi x10, x0, 5

jal x1, fact #addr=2000

x5	4
x10	4

x2 (sp)	984
x1 (ra)	2004

936	
944	
952	
960	
968	
976	
984	5
992	2004
1000	

# Example

fact:

addi sp,sp,-16  Save return address and n on stack

sd x1,8(sp)

sd x10,0(sp)

addi x5,x10,-1  $x5 = n - 1$

bge x5,x0,L1 if  $n \geq 0$ , go to L1

addi x10,x0,1 Else, set return value to 1

addi sp,sp,16 Pop stack, don't bother restoring values

jr x1

L1: addi x10,x10,-1

jal x1,fact

addi x6,x10,0

ld x10,0(sp)

ld x1,8(sp)

addi sp,sp,16

mul x10,x10,x6

jr x1

Return

$n = n - 1$

*/\*Address = 5000\*/*

move result of fact( $n - 1$ ) to x6

Restore caller's n

Restore caller's return address

Pop stack

return  $n * \text{fact}(n-1)$

return

x5	4
x10	4

x2 (sp)	984
x1 (ra)	5004

936	
944	
952	
960	
968	
976	
984	5
992	2004
1000	

main:

addi x10, x0, 5

jal x1, fact *#addr=2000*

# Example

fact:

addi sp,sp,-16

Save return address and n on stack

sd x1,8(sp)

sd x10,0(sp)

addi x5,x10,-1

x5 = n - 1

bge x5,x0,L1

if n >= 0, go to L1

addi x10,x0,1

Else, set return value to 1

addi sp,sp,16

Pop stack, don't bother restoring values

jr x1

Return

L1: addi x10,x10,-1

n = n - 1

jal x1,fact

/\*Address = 5000\*/

addi x6,x10,0

move result of fact(n - 1) to x6

ld x10,0(sp)

Restore caller's n

ld x1,8(sp)

Restore caller's return address

addi sp,sp,16

Pop stack

mul x10,x10,x6

return n \* fact(n-1)

jr x1

return

main:

addi x10, x0, 5

jal x1, fact #addr=2000

x5	4
x10	4

x2 (sp)	968
x1 (ra)	5004

936	
944	
952	
960	
968	4
976	5004
984	5
992	2004
1000	

# Example

fact:

addi sp,sp,-16

Save return address and n on stack

sd x1,8(sp)

sd x10,0(sp)

addi x5,x10,-1

$x5 = n - 1$

bge x5,x0,L1

if  $n \geq 0$ , go to L1

addi x10,x0,1

Else, set return value to 1

addi sp,sp,16

Pop stack, don't bother restoring values

jr x1

Return

L1: addi x10,x10,-1

$n = n - 1$

jal x1,fact

/\*Address = 5000\*/

addi x6,x10,0

move result of fact( $n - 1$ ) to x6

ld x10,0(sp)

Restore caller's n

ld x1,8(sp)

Restore caller's return address

addi sp,sp,16

Pop stack

mul x10,x10,x6

return  $n * \text{fact}(n-1)$

jr x1

return

main:

addi x10, x0, 5

jal x1, fact #addr=2000

x5	3
x10	3

x2 (sp)	968
x1 (ra)	5004

936	
944	
952	
960	
968	4
976	5004
984	5
992	2004
1000	

# Example

fact:

addi sp,sp,-16

Save return address and n on stack

sd x1,8(sp)

sd x10,0(sp)

addi x5,x10,-1

$x5 = n - 1$

bge x5,x0,L1

if  $n \geq 0$ , go to L1

addi x10,x0,1

Else, set return value to 1

addi sp,sp,16

Pop stack, don't bother restoring values

jr x1

Return

L1: addi x10,x10,-1

$n = n - 1$

jal x1,fact

*/\*Address = 5000\*/*

addi x6,x10,0

move result of fact( $n - 1$ ) to x6

ld x10,0(sp)

Restore caller's n

ld x1,8(sp)

Restore caller's return address

addi sp,sp,16

Pop stack

mul x10,x10,x6

return  $n * \text{fact}(n-1)$

jr x1

return

main:

addi x10, x0, 5

jal x1, fact *#addr=2000*

x5	2
x10	3

x2 (sp)	952
x1 (ra)	5004

936	
944	
952	3
960	5004
968	4
976	5004
984	5
992	2004
1000	

# Example

fact:

addi sp,sp,-16

Save return address and n on stack

sd x1,8(sp)

sd x10,0(sp)

addi x5,x10,-1

$x5 = n - 1$

bge x5,x0,L1

if  $n \geq 0$ , go to L1

addi x10,x0,1

Else, set return value to 1

addi sp,sp,16

Pop stack, don't bother restoring values

jr x1

Return

L1: addi x10,x10,-1

$n = n - 1$

jal x1,fact

*/\*Address = 5000\*/*

addi x6,x10,0

move result of fact( $n - 1$ ) to x6

ld x10,0(sp)

Restore caller's n

ld x1,8(sp)

Restore caller's return address

addi sp,sp,16

Pop stack

mul x10,x10,x6

return  $n * \text{fact}(n-1)$

jr x1

return

main:

addi x10, x0, 5

jal x1, fact *#addr=2000*

x5	2
x10	2

x2 (sp)	952
x1 (ra)	5004

936	
944	
952	3
960	5004
968	4
976	5004
984	5
992	2004
1000	

# Example

fact:

addi sp,sp,-16

Save return address and n on stack

sd x1,8(sp)

sd x10,0(sp)

addi x5,x10,-1

$x5 = n - 1$

bge x5,x0,L1

if  $n \geq 0$ , go to L1

addi x10,x0,1

Else, set return value to 1

addi sp,sp,16

Pop stack, don't bother restoring values

jr x1

Return

L1: addi x10,x10,-1

$n = n - 1$

jal x1,fact

*/\*Address = 5000\*/*

addi x6,x10,0

move result of fact( $n - 1$ ) to x6

ld x10,0(sp)

Restore caller's n

ld x1,8(sp)

Restore caller's return address

addi sp,sp,16

Pop stack

mul x10,x10,x6

return  $n * \text{fact}(n-1)$

jr x1

return

main:

addi x10, x0, 5

jal x1, fact *#addr=2000*

x5	1
x10	2

x2 (sp)	936
x1 (ra)	5004

920	
928	
936	2
944	5004
952	3
960	5004
968	4
976	5004
984	5
992	2004
1000	

# Example

fact:

addi sp,sp,-16

Save return address and n on stack

sd x1,8(sp)

sd x10,0(sp)

addi x5,x10,-1

$x5 = n - 1$

bge x5,x0,L1

if  $n \geq 0$ , go to L1

addi x10,x0,1

Else, set return value to 1

addi sp,sp,16

Pop stack, don't bother restoring values

jr x1

Return

L1: addi x10,x10,-1

$n = n - 1$

jal x1,fact

/\*Address = 5000\*/

addi x6,x10,0

move result of fact( $n - 1$ ) to x6

ld x10,0(sp)

Restore caller's n

ld x1,8(sp)

Restore caller's return address

addi sp,sp,16

Pop stack

mul x10,x10,x6

return  $n * \text{fact}(n-1)$

jr x1

return

main:

addi x10, x0, 5

jal x1, fact #addr=2000

x5	1
x10	1

x2 (sp)	936
x1 (ra)	5004

920	
928	
936	2
944	5004
952	3
960	5004
968	4
976	5004
984	5
992	2004
1000	



# Example

fact:

addi sp,sp,-16

Save return address and n on stack

sd x1,8(sp)

sd x10,0(sp)

addi x5,x10,-1

$x5 = n - 1$

bge x5,x0,L1

if  $n \geq 0$ , go to L1

addi x10,x0,1

Else, set return value to 1

addi sp,sp,16

Pop stack, don't bother restoring values

jr x1

Return

L1: addi x10,x10,-1

$n = n - 1$

jal x1,fact

*/\*Address = 5000\*/*

addi x6,x10,0

move result of fact( $n - 1$ ) to x6

ld x10,0(sp)

Restore caller's n

ld x1,8(sp)

Restore caller's return address

addi sp,sp,16

Pop stack

mul x10,x10,x6

return  $n * \text{fact}(n-1)$

jr x1

return

main:

addi x10, x0, 5

jal x1, fact *#addr=2000*

x5	0
x10	1

x2 (sp)	920
x1 (ra)	5004

920	1
928	5004
936	2
944	5004
952	3
960	5004
968	4
976	5004
984	5
992	2004
1000	

# Example

fact:

addi sp,sp,-16

Save return address and n on stack

sd x1,8(sp)

sd x10,0(sp)

addi x5,x10,-1

$x5 = n - 1$

bge x5,x0,L1

if  $n \geq 0$ , go to L1

addi x10,x0,1

Else, set return value to 1

addi sp,sp,16

Pop stack, don't bother restoring values

jr x1

Return

L1: addi x10,x10,-1

$n = n - 1$

jal x1,fact

/\*Address = 5000\*/

addi x6,x10,0

move result of fact( $n - 1$ ) to x6

ld x10,0(sp)

Restore caller's n

ld x1,8(sp)

Restore caller's return address

addi sp,sp,16

Pop stack

mul x10,x10,x6

return  $n * \text{fact}(n-1)$

jr x1

return

main:

addi x10, x0, 5

jal x1, fact #addr=2000

x5	0
x10	1

x2 (sp)	920
x1 (ra)	5004

920	1
928	5004
936	2
944	5004
952	3
960	5004
968	4
976	5004
984	5
992	2004
1000	

# Example

fact:

addi sp,sp,-16

Save return address and n on stack

sd x1,8(sp)

sd x10,0(sp)

addi x5,x10,-1

$x5 = n - 1$

bge x5,x0,L1

if  $n \geq 0$ , go to L1

addi x10,x0,1

Else, set return value to 1

addi sp,sp,16

Pop stack, don't bother restoring values

jr x1

Return

L1: addi x10,x10,-1

$n = n - 1$

jal x1,fact

/\*Address = 5000\*/

addi x6,x10,0

move result of fact( $n - 1$ ) to x6

ld x10,0(sp)

Restore caller's n

ld x1,8(sp)

Restore caller's return address

addi sp,sp,16

Pop stack

mul x10,x10,x6

return  $n * \text{fact}(n-1)$

jr x1

return

main:

addi x10, x0, 5

jal x1, fact #addr=2000

x5	0
x10	0

x2 (sp)	920
x1 (ra)	5004

920	1
928	5004
936	2
944	5004
952	3
960	5004
968	4
976	5004
984	5
992	2004
1000	

# Example

fact:

addi sp,sp,-16

Save return address and n on stack

sd x1,8(sp)

sd x10,0(sp)

addi x5,x10,-1

x5 = n - 1

bge x5,x0,L1

if n >= 0, go to L1

addi x10,x0,1

Else, set return value to 1

addi sp,sp,16

Pop stack, don't bother restoring values

jr x1

Return

L1: addi x10,x10,-1

n = n - 1

jal x1,fact

/\*Address = 5000\*/

addi x6,x10,0

move result of fact(n - 1) to x6

ld x10,0(sp)

Restore caller's n

ld x1,8(sp)

Restore caller's return address

addi sp,sp,16

Pop stack

mul x10,x10,x6

return n \* fact(n-1)

jr x1

return

main:

addi x10, x0, 5

jal x1, fact #addr=2000

x5	0
x10	0

x2 (sp)	904
x1 (ra)	5004

904	0
912	5004
920	1
928	5004
936	2
944	5004
952	3
960	5004
968	4
976	5004
984	5
992	2004
1000	

# Example

fact:

addi sp,sp,-16

Save return address and n on stack

sd x1,8(sp)

sd x10,0(sp)

addi x5,x10,-1



x5 = n - 1

bge x5,x0,L1

if n >= 0, go to L1

addi x10,x0,1

Else, set return value to 1

addi sp,sp,16

Pop stack, don't bother restoring values

jr x1

Return

L1: addi x10,x10,-1

n = n - 1

jal x1,fact

/\*Address = 5000\*/

addi x6,x10,0

move result of fact(n - 1) to x6

ld x10,0(sp)

Restore caller's n

ld x1,8(sp)

Restore caller's return address

addi sp,sp,16

Pop stack

mul x10,x10,x6

return n \* fact(n-1)

jr x1

return

main:

addi x10, x0, 5

jal x1, fact #addr=2000

x5	-1
x10	0

x2 (sp)	904
x1 (ra)	5004

904	0
912	5004
920	1
928	5004
936	2
944	5004
952	3
960	5004
968	4
976	5004
984	5
992	2004
1000	

# Example

fact:

addi sp,sp,-16

Save return address and n on stack

sd x1,8(sp)

sd x10,0(sp)

addi x5,x10,-1

$x5 = n - 1$

bge x5,x0,L1

if  $n \geq 0$ , go to L1

addi x10,x0,1

Else, set return value to 1

addi sp,sp,16

Pop stack, don't bother restoring values

jr x1

Return

L1: addi x10,x10,-1

$n = n - 1$

jal x1,fact

*/\*Address = 5000\*/*

addi x6,x10,0

move result of fact( $n - 1$ ) to x6

ld x10,0(sp)

Restore caller's n

ld x1,8(sp)

Restore caller's return address

addi sp,sp,16

Pop stack

mul x10,x10,x6

return  $n * \text{fact}(n-1)$

jr x1

return

main:

addi x10, x0, 5

jal x1, fact *#addr=2000*

x5	-1
x10	1

x2 (sp)	920
x1 (ra)	5004

904	<del>0</del>
912	<del>5004</del>
920	1
928	5004
936	2
944	5004
952	3
960	5004
968	4
976	5004
984	5
992	2004
1000	

# Example

fact:

addi sp,sp,-16

Save return address and n on stack

sd x1,8(sp)

sd x10,0(sp)

addi x5,x10,-1

$x5 = n - 1$

bge x5,x0,L1

if  $n \geq 0$ , go to L1

addi x10,x0,1

Else, set return value to 1

addi sp,sp,16

Pop stack, don't bother restoring values

jr x1

Return

L1: addi x10,x10,-1

$n = n - 1$

jal x1,fact

*/\*Address = 5000\*/*

addi x6,x10,0



move result of fact( $n - 1$ ) to x6

ld x10,0(sp)

Restore caller's n

ld x1,8(sp)

Restore caller's return address

addi sp,sp,16

Pop stack

mul x10,x10,x6

return  $n * \text{fact}(n-1)$

jr x1

return

main:

addi x10, x0, 5

jal x1, fact *#addr=2000*

x6	
x5	-1
x10	1

x2 (sp)	920
x1 (ra)	5004

920	1
928	5004
936	2
944	5004
952	3
960	5004
968	4
976	5004
984	5
992	2004
1000	

# Example

fact:

addi sp,sp,-16

Save return address and n on stack

sd x1,8(sp)

sd x10,0(sp)

addi x5,x10,-1

$x5 = n - 1$

bge x5,x0,L1

if  $n \geq 0$ , go to L1

addi x10,x0,1

Else, set return value to 1

addi sp,sp,16

Pop stack, don't bother restoring values

jr x1

Return

L1: addi x10,x10,-1

$n = n - 1$

jal x1,fact

*/\*Address = 5000\*/*

addi x6,x10,0

move result of fact( $n - 1$ ) to x6

ld x10,0(sp)

Restore caller's n

ld x1,8(sp)

Restore caller's return address

addi sp,sp,16

Pop stack

mul x10,x10,x6

return  $n * \text{fact}(n-1)$

jr x1

return

main:

addi x10, x0, 5

jal x1, fact *#addr=2000*

x6	1
x5	-1
x10	1

x2 (sp)	920
x1 (ra)	5004

920	1
928	5004
936	2
944	5004
952	3
960	5004
968	4
976	5004
984	5
992	2004
1000	



# Example

fact:

```
addi sp, sp, -16
```

Save return address and n on stack

```
sd x1, 8(sp)
```

```
sd x10, 0(sp)
```

```
addi x5, x10, -1
```

$x5 = n - 1$

```
bge x5, x0, L1
```

if  $n \geq 0$ , go to L1

```
addi x10, x0, 1
```

Else, set return value to 1

```
addi sp, sp, 16
```

Pop stack, don't bother restoring values

```
jr x1
```

Return

```
L1: addi x10, x10, -1
```

$n = n - 1$

```
jal x1, fact
```

*/\*Address = 5000\*/*

```
addi x6, x10, 0
```



move result of fact( $n - 1$ ) to x6

```
ld x10, 0(sp)
```

Restore caller's n

```
ld x1, 8(sp)
```

Restore caller's return address

```
addi sp, sp, 16
```

Pop stack

```
mul x10, x10, x6
```

return  $n * \text{fact}(n-1)$

```
jr x1
```

return

main:

```
addi x10, x0, 5
```

```
jal x1, fact #addr=2000
```

x6	1
x5	0
x10	1

x2 (sp)	936
x1 (ra)	5004

920	<del>1</del>
928	<del>5004</del>
936	2
944	5004
952	3
960	5004
968	4
976	5004
984	5
992	2004
1000	

# Example

fact:

```
addi sp, sp, -16
```

Save return address and n on stack

```
sd x1, 8(sp)
```

```
sd x10, 0(sp)
```

```
addi x5, x10, -1
```

$x5 = n - 1$

```
bge x5, x0, L1
```

if  $n \geq 0$ , go to L1

```
addi x10, x0, 1
```

Else, set return value to 1

```
addi sp, sp, 16
```

Pop stack, don't bother restoring values

```
jr x1
```

Return

```
L1: addi x10, x10, -1
```

$n = n - 1$

```
jal x1, fact
```

*/\*Address = 5000\*/*

```
addi x6, x10, 0
```

move result of fact( $n - 1$ ) to x6

```
ld x10, 0(sp)
```

Restore caller's n

```
ld x1, 8(sp)
```

Restore caller's return address

```
addi sp, sp, 16
```

Pop stack

```
mul x10, x10, x6
```

return  $n * \text{fact}(n-1)$

```
jr x1
```

return

main:

```
addi x10, x0, 5
```

```
jal x1, fact #addr=2000
```

x6	1
x5	0
x10	1

x2 (sp)	936
x1 (ra)	5004

920	
928	
936	2
944	5004
952	3
960	5004
968	4
976	5004
984	5
992	2004
1000	

# Example

fact:

addi sp,sp,-16

Save return address and n on stack

sd x1,8(sp)

sd x10,0(sp)

addi x5,x10,-1

$x5 = n - 1$

bge x5,x0,L1

if  $n \geq 0$ , go to L1

addi x10,x0,1

Else, set return value to 1

addi sp,sp,16

Pop stack, don't bother restoring values

jr x1

Return

L1: addi x10,x10,-1

$n = n - 1$

jal x1,fact

*/\*Address = 5000\*/*

addi x6,x10,0

move result of fact( $n - 1$ ) to x6

ld x10,0(sp)

Restore caller's n

ld x1,8(sp)

Restore caller's return address

addi sp,sp,16

Pop stack

mul x10,x10,x6

return  $n * \text{fact}(n-1)$

jr x1

return

main:

addi x10, x0, 5

jal x1, fact *#addr=2000*

x6	1
x5	0
x10	2

x2 (sp)	952
x1 (ra)	5004

920	
928	
936	
944	<del>2</del>
952	<del>5004</del>
960	3
968	5004
976	4
984	5004
992	5
1000	2004



# Example

fact:

```
addi sp, sp, -16
```

Save return address and n on stack

```
sd x1, 8(sp)
```

```
sd x10, 0(sp)
```

```
addi x5, x10, -1
```

$x5 = n - 1$

```
bge x5, x0, L1
```

if  $n \geq 0$ , go to L1

```
addi x10, x0, 1
```

Else, set return value to 1

```
addi sp, sp, 16
```

Pop stack, don't bother restoring values

```
jr x1
```

Return

```
L1: addi x10, x10, -1
```

$n = n - 1$

```
jal x1, fact
```

*/\*Address = 5000\*/*

```
addi x6, x10, 0
```

move result of  $\text{fact}(n - 1)$  to x6

```
ld x10, 0(sp)
```



Restore caller's n

```
ld x1, 8(sp)
```

Restore caller's return address

```
addi sp, sp, 16
```

Pop stack

```
mul x10, x10, x6
```

return  $n * \text{fact}(n-1)$

```
jr x1
```

return

main:

```
addi x10, x0, 5
```

```
jal x1, fact #addr=2000
```

x6	2
x5	0
x10	2

x2 (sp)	952
x1 (ra)	5004

920	
928	
936	
944	
952	3
960	5004
968	4
976	5004
984	5
992	2004
1000	

# Example

fact:

```
addi sp, sp, -16
```

Save return address and n on stack

```
sd x1, 8(sp)
```

```
sd x10, 0(sp)
```

```
addi x5, x10, -1
```

$x5 = n - 1$

```
bge x5, x0, L1
```

if  $n \geq 0$ , go to L1

```
addi x10, x0, 1
```

Else, set return value to 1

```
addi sp, sp, 16
```

Pop stack, don't bother restoring values

```
jr x1
```

Return

```
L1: addi x10, x10, -1
```

$n = n - 1$

```
jal x1, fact
```

*/\*Address = 5000\*/*

```
addi x6, x10, 0
```

move result of  $\text{fact}(n - 1)$  to  $x6$

```
ld x10, 0(sp)
```

Restore caller's  $n$

```
ld x1, 8(sp)
```

Restore caller's return address

```
addi sp, sp, 16
```

Pop stack

```
mul x10, x10, x6
```

return  $n * \text{fact}(n-1)$

```
jr x1
```

return

main:

```
addi x10, x0, 5
```

```
jal x1, fact #addr=2000
```

x6	2
x5	0
x10	6

x2 (sp)	968
x1 (ra)	5004

920	
928	
936	
944	
952	<del>3</del>
960	<del>5004</del>
968	4
976	5004
984	5
992	2004
1000	



# Example

fact:

addi sp,sp,-16

Save return address and n on stack

sd x1,8(sp)

sd x10,0(sp)

addi x5,x10,-1

$x5 = n - 1$

bge x5,x0,L1

if  $n \geq 0$ , go to L1

addi x10,x0,1

Else, set return value to 1

addi sp,sp,16

Pop stack, don't bother restoring values

jr x1

Return

L1: addi x10,x10,-1

$n = n - 1$

jal x1,fact

*/\*Address = 5000\*/*

addi x6,x10,0

move result of fact( $n - 1$ ) to x6

ld x10,0(sp)

Restore caller's n

ld x1,8(sp)

Restore caller's return address

addi sp,sp,16

Pop stack

mul x10,x10,x6

return  $n * \text{fact}(n-1)$

jr x1

return

main:

addi x10, x0, 5

jal x1, fact *#addr=2000*

x6	2
x5	0
x10	6

x2 (sp)	968
x1 (ra)	5004

920	
928	
936	
944	
952	
960	
968	4
976	5004
984	5
992	2004
1000	

# Example

fact:

```
addi sp, sp, -16
```

Save return address and n on stack

```
sd x1, 8(sp)
```

```
sd x10, 0(sp)
```

```
addi x5, x10, -1
```

$x5 = n - 1$

```
bge x5, x0, L1
```

if  $n \geq 0$ , go to L1

```
addi x10, x0, 1
```

Else, set return value to 1

```
addi sp, sp, 16
```

Pop stack, don't bother restoring values

```
jr x1
```

Return

```
L1: addi x10, x10, -1
```

$n = n - 1$

```
jal x1, fact
```

*/\*Address = 5000\*/*

```
addi x6, x10, 0
```

move result of  $\text{fact}(n - 1)$  to  $x6$

```
ld x10, 0(sp)
```

Restore caller's  $n$

```
ld x1, 8(sp)
```

Restore caller's return address

```
addi sp, sp, 16
```

Pop stack

```
mul x10, x10, x6
```

return  $n * \text{fact}(n-1)$

```
jr x1
```

return

main:

```
addi x10, x0, 5
```

```
jal x1, fact #addr=2000
```

x6	6
x5	0
x10	24

x2 (sp)	984
x1 (ra)	5004

920	
928	
936	
944	
952	
960	
968	<del>4</del>
976	<del>5004</del>
984	5
992	
1000	2004

# Example

fact:

addi sp,sp,-16

Save return address and n on stack

sd x1,8(sp)

sd x10,0(sp)

addi x5,x10,-1

$x5 = n - 1$

bge x5,x0,L1

if  $n \geq 0$ , go to L1

addi x10,x0,1

Else, set return value to 1

addi sp,sp,16

Pop stack, don't bother restoring values

jr x1

Return

L1: addi x10,x10,-1

$n = n - 1$

jal x1,fact

*/\*Address = 5000\*/*

addi x6,x10,0

move result of fact( $n - 1$ ) to x6

ld x10,0(sp)

Restore caller's n

ld x1,8(sp)

Restore caller's return address

addi sp,sp,16

Pop stack

mul x10,x10,x6

return  $n * \text{fact}(n-1)$

jr x1

return

main:

addi x10, x0, 5

jal x1, fact *#addr=2000*

x6	6
x5	0
x10	24

x2 (sp)	984
x1 (ra)	5004

920	
928	
936	
944	
952	
960	
968	
976	
984	5
992	2004
1000	



# Example

fact:

addi sp,sp,-16

Save return address and n on stack

sd x1,8(sp)

sd x10,0(sp)

addi x5,x10,-1

$x5 = n - 1$

bge x5,x0,L1

if  $n \geq 0$ , go to L1

addi x10,x0,1

Else, set return value to 1

addi sp,sp,16

Pop stack, don't bother restoring values

jr x1

Return

L1: addi x10,x10,-1

$n = n - 1$

jal x1,fact

*/\*Address = 5000\*/*

addi x6,x10,0

move result of fact( $n - 1$ ) to x6

ld x10,0(sp)

Restore caller's n

ld x1,8(sp)

Restore caller's return address

addi sp,sp,16

Pop stack

mul x10,x10,x6

return  $n * \text{fact}(n-1)$

jr x1

return

main:

addi x10, x0, 5

jal x1, fact *#addr=2000*

x6	24
x5	0
x10	5

x2 (sp)	984
x1 (ra)	2004

920	
928	
936	
944	
952	
960	
968	
976	
984	5
992	2004
1000	

# Example

fact:

```
addi sp, sp, -16
```

Save return address and n on stack

```
sd x1, 8(sp)
```

```
sd x10, 0(sp)
```

```
addi x5, x10, -1
```

$x5 = n - 1$

```
bge x5, x0, L1
```

if  $n \geq 0$ , go to L1

```
addi x10, x0, 1
```

Else, set return value to 1

```
addi sp, sp, 16
```

Pop stack, don't bother restoring values

```
jr x1
```

Return

```
L1: addi x10, x10, -1
```

$n = n - 1$

```
jal x1, fact
```

*/\*Address = 5000\*/*

```
addi x6, x10, 0
```

move result of  $\text{fact}(n - 1)$  to  $x6$

```
ld x10, 0(sp)
```

Restore caller's  $n$

```
ld x1, 8(sp)
```

Restore caller's return address

```
addi sp, sp, 16
```

Pop stack

```
mul x10, x10, x6
```

return  $n * \text{fact}(n-1)$

```
jr x1
```

return

main:

```
addi x10, x0, 5
```

```
jal x1, fact #addr=2000
```

x6	24
x5	0
x10	120

x2 (sp)	1000
x1 (ra)	2004

920	
928	
936	
944	
952	
960	
968	
976	
984	<del>5</del>
992	<del>2004</del>
1000	

# Example

fact:

addi sp,sp,-16

Save return address and n on stack

sd x1,8(sp)

sd x10,0(sp)

addi x5,x10,-1

$x5 = n - 1$

bge x5,x0,L1

if  $n \geq 0$ , go to L1

addi x10,x0,1

Else, set return value to 1

addi sp,sp,16

Pop stack, don't bother restoring values

jr x1

Return

L1: addi x10,x10,-1

$n = n - 1$

jal x1,fact

*/\*Address = 5000\*/*

addi x6,x10,0

move result of fact( $n - 1$ ) to x6

ld x10,0(sp)

Restore caller's n

ld x1,8(sp)

Restore caller's return address

addi sp,sp,16

Pop stack

mul x10,x10,x6

return  $n * \text{fact}(n-1)$

jr x1

return

main:

addi x10, x0, 5

jal x1, fact *#addr=2000*



x6	24
x5	0
x10	120

x2 (sp)	1000
x1 (ra)	2004

920	
928	
936	
944	
952	
960	
968	
976	
984	
992	
1000	