



	5056				
	5052			\$ra	8004
sp→	5048			\$a0	4
	5044			\$t0	
	5040			\$v0	
	5036				
	5032				
	5028				
	5024				
	5020				
	5016				
	5012				
	5008				
	5004				
	5000				
	8000	jal fact			
	8004				
	...				
pc→	10000	fact: addi \$sp, \$sp, -8	#adjust stack for 2 items		
	10004	sw \$ra, 4(\$sp)	#save the return address		
	10008	sw \$a0, 0(\$sp)	#save the argument n		
	10012	slti \$t0,\$a0,1	#test for n < 1		
	10016	beq \$t0,\$zero,L1	#if n >= 1, go to L1		
	10020	addi \$v0,\$zero,1	#return 1		
	10024	addi \$sp,\$sp,8	#pop 2 items off stack		
	10028	jr \$ra	#return to caller		
	10032	L1: addi \$a0,\$a0,-1	#n >= 1: argument gets (n - 1)		
	10036	jal fact	#call fact with (n - 1)		
	10040	lw \$a0, 0(\$sp)	#return from jal: restore argument n		
	10044	lw \$ra, 4(\$sp)	#restore the return address		
	10048	addi \$sp, \$sp, 8	#adjust stack pointer to pop 2 items		
	10052	mul \$v0,\$a0,\$v0	#return n * fact (n - 1)		
	10056	jr \$ra	#return to the caller		

	5056				
	5052	\$ra	8004	\$ra	8004
<b>sp→</b>	5048			\$a0	4
	5044			\$t0	
	5040			\$v0	
	5036				
	5032				
	5028				
	5024				
	5020				
	5016				
	5012				
	5008				
	5004				
	5000				
	8000	jal fact			
	8004				
	...				
	10000	fact: addi \$sp, \$sp, -8	#adjust stack for 2 items		
<b>pc→</b>	10004	sw \$ra, 4(\$sp)	#save the return address		
	10008	sw \$a0, 0(\$sp)	#save the argument n		
	10012	slti \$t0,\$a0,1	#test for n < 1		
	10016	beq \$t0,\$zero,L1	#if n >= 1, go to L1		
	10020	addi \$v0,\$zero,1	#return 1		
	10024	addi \$sp,\$sp,8	#pop 2 items off stack		
	10028	jr \$ra	#return to caller		
	10032	L1: addi \$a0,\$a0,-1	#n >= 1: argument gets (n - 1)		
	10036	jal fact	#call fact with (n - 1)		
	10040	lw \$a0, 0(\$sp)	#return from jal: restore argument n		
	10044	lw \$ra, 4(\$sp)	#restore the return address		
	10048	addi \$sp, \$sp, 8	#adjust stack pointer to pop 2 items		
	10052	mul \$v0,\$a0,\$v0	#return n * fact (n - 1)		
	10056	jr \$ra	#return to the caller		

	5056				
	5052	\$ra	8004	\$ra	8004
<b>sp</b> →	5048	\$a0	4	\$a0	4
	5044			\$t0	
	5040			\$v0	
	5036				
	5032				
	5028				
	5024				
	5020				
	5016				
	5012				
	5008				
	5004				
	5000				
	8000	jal fact			
	8004				
	...				
	10000	fact: addi \$sp, \$sp, -8	#adjust stack for 2 items		
	10004	sw \$ra, 4(\$sp)	#save the return address		
<b>pc</b> →	10008	sw \$a0, 0(\$sp)	#save the argument n		
	10012	slti \$t0,\$a0,1	#test for n < 1		
	10016	beq \$t0,\$zero,L1	#if n >= 1, go to L1		
	10020	addi \$v0,\$zero,1	#return 1		
	10024	addi \$sp,\$sp,8	#pop 2 items off stack		
	10028	jr \$ra	#return to caller		
	10032	L1: addi \$a0,\$a0,-1	#n >= 1: argument gets (n - 1)		
	10036	jal fact	#call fact with (n - 1)		
	10040	lw \$a0, 0(\$sp)	#return from jal: restore argument n		
	10044	lw \$ra, 4(\$sp)	#restore the return address		
	10048	addi \$sp, \$sp, 8	#adjust stack pointer to pop 2 items		
	10052	mul \$v0,\$a0,\$v0	#return n * fact (n - 1)		
	10056	jr \$ra	#return to the caller		

	5056				
	5052	\$ra	8004	\$ra	8004
<b>sp</b> →	5048	\$a0	4	\$a0	4
	5044			\$t0	0
	5040			\$v0	
	5036				
	5032				
	5028				
	5024				
	5020				
	5016				
	5012				
	5008				
	5004				
	5000				
	8000	jal fact			
	8004				
	...				
	10000	fact: addi \$sp, \$sp, -8	#adjust stack for 2 items		
	10004	sw \$ra, 4(\$sp)	#save the return address		
	10008	sw \$a0, 0(\$sp)	#save the argument n		
<b>pc</b> →	10012	slti \$t0,\$a0,1	#test for n < 1		
	10016	beq \$t0,\$zero,L1	#if n >= 1, go to L1		
	10020	addi \$v0,\$zero,1	#return 1		
	10024	addi \$sp,\$sp,8	#pop 2 items off stack		
	10028	jr \$ra	#return to caller		
	10032	L1: addi \$a0,\$a0,-1	#n >= 1: argument gets (n - 1)		
	10036	jal fact	#call fact with (n - 1)		
	10040	lw \$a0, 0(\$sp)	#return from jal: restore argument n		
	10044	lw \$ra, 4(\$sp)	#restore the return address		
	10048	addi \$sp, \$sp, 8	#adjust stack pointer to pop 2 items		
	10052	mul \$v0,\$a0,\$v0	#return n * fact (n - 1)		
	10056	jr \$ra	#return to the caller		

	5056				
	5052	\$ra	8004	\$ra	8004
<b>sp→</b>	5048	\$a0	4	\$a0	4
	5044			\$t0	0
	5040			\$v0	
	5036				
	5032				
	5028				
	5024				
	5020				
	5016				
	5012				
	5008				
	5004				
	5000				
	8000	jal fact			
	8004				
	...				
	10000	fact: addi \$sp, \$sp, -8	#adjust stack for 2 items		
	10004	sw \$ra, 4(\$sp)	#save the return address		
	10008	sw \$a0, 0(\$sp)	#save the argument n		
	10012	slti \$t0,\$a0,1	#test for n < 1		
<b>pc→</b>	10016	beq \$t0,\$zero,L1	#if n >= 1, go to L1		
	10020	addi \$v0,\$zero,1	#return 1		
	10024	addi \$sp,\$sp,8	#pop 2 items off stack		
	10028	jr \$ra	#return to caller		
	10032	L1: addi \$a0,\$a0,-1	#n >= 1: argument gets (n - 1)		
	10036	jal fact	#call fact with (n - 1)		
	10040	lw \$a0, 0(\$sp)	#return from jal: restore argument n		
	10044	lw \$ra, 4(\$sp)	#restore the return address		
	10048	addi \$sp, \$sp, 8	#adjust stack pointer to pop 2 items		
	10052	mul \$v0,\$a0,\$v0	#return n * fact (n - 1)		
	10056	jr \$ra	#return to the caller		

	5056				
	5052	\$ra	8004	\$ra	8004
<b>sp→</b>	5048	\$a0	4	\$a0	3
	5044			\$t0	0
	5040			\$v0	
	5036				
	5032				
	5028				
	5024				
	5020				
	5016				
	5012				
	5008				
	5004				
	5000				
	8000	jal fact			
	8004				
	...				
	10000	fact: addi \$sp, \$sp, -8	#adjust stack for 2 items		
	10004	sw \$ra, 4(\$sp)	#save the return address		
	10008	sw \$a0, 0(\$sp)	#save the argument n		
	10012	slti \$t0,\$a0,1	#test for n < 1		
	10016	beq \$t0,\$zero,L1	#if n >= 1, go to L1		
	10020	addi \$v0,\$zero,1	#return 1		
	10024	addi \$sp,\$sp,8	#pop 2 items off stack		
	10028	jr \$ra	#return to caller		
<b>pc→</b>	10032	L1: addi \$a0,\$a0,-1	#n >= 1: argument gets (n - 1)		
	10036	jal fact	#call fact with (n - 1)		
	10040	lw \$a0, 0(\$sp)	#return from jal: restore argument n		
	10044	lw \$ra, 4(\$sp)	#restore the return address		
	10048	addi \$sp, \$sp, 8	#adjust stack pointer to pop 2 items		
	10052	mul \$v0,\$a0,\$v0	#return n * fact (n - 1)		
	10056	jr \$ra	#return to the caller		

	5056				
	5052	\$ra	8004	\$ra	10040
<b>sp→</b>	5048	\$a0	4	\$a0	3
	5044			\$t0	0
	5040			\$v0	
	5036				
	5032				
	5028				
	5024				
	5020				
	5016				
	5012				
	5008				
	5004				
	5000				
	8000	jal fact			
	8004				
	...				
	10000	fact: addi \$sp, \$sp, -8	#adjust stack for 2 items		
	10004	sw \$ra, 4(\$sp)	#save the return address		
	10008	sw \$a0, 0(\$sp)	#save the argument n		
	10012	slti \$t0,\$a0,1	#test for n < 1		
	10016	beq \$t0,\$zero,L1	#if n >= 1, go to L1		
	10020	addi \$v0,\$zero,1	#return 1		
	10024	addi \$sp,\$sp,8	#pop 2 items off stack		
	10028	jr \$ra	#return to caller		
	10032	L1: addi \$a0,\$a0,-1	#n >= 1: argument gets (n - 1)		
<b>pc→</b>	10036	jal fact	#call fact with (n - 1)		
	10040	lw \$a0, 0(\$sp)	#return from jal: restore argument n		
	10044	lw \$ra, 4(\$sp)	#restore the return address		
	10048	addi \$sp, \$sp, 8	#adjust stack pointer to pop 2 items		
	10052	mul \$v0,\$a0,\$v0	#return n * fact (n - 1)		
	10056	jr \$ra	#return to the caller		



	5056				
	5052	\$ra	8004	\$ra	10040
	5048	\$a0	4	\$a0	3
	5044			\$t0	0
<b>sp→</b>	5040			\$v0	
	5036				
	5032				
	5028				
	5024				
	5020				
	5016				
	5012				
	5008				
	5004				
	5000				
	8000	jal fact			
	8004				
	...				
<b>pc→</b>	10000	fact: addi \$sp, \$sp, -8	#adjust stack for 2 items		
	10004	sw \$ra, 4(\$sp)	#save the return address		
	10008	sw \$a0, 0(\$sp)	#save the argument n		
	10012	slti \$t0,\$a0,1	#test for n < 1		
	10016	beq \$t0,\$zero,L1	#if n >= 1, go to L1		
	10020	addi \$v0,\$zero,1	#return 1		
	10024	addi \$sp,\$sp,8	#pop 2 items off stack		
	10028	jr \$ra	#return to caller		
	10032	L1: addi \$a0,\$a0,-1	#n >= 1: argument gets (n - 1)		
	10036	jal fact	#call fact with (n - 1)		
	10040	lw \$a0, 0(\$sp)	#return from jal: restore argument n		
	10044	lw \$ra, 4(\$sp)	#restore the return address		
	10048	addi \$sp, \$sp, 8	#adjust stack pointer to pop 2 items		
	10052	mul \$v0,\$a0,\$v0	#return n * fact (n - 1)		
	10056	jr \$ra	#return to the caller		

	5056				
	5052	\$ra	8004	\$ra	10040
	5048	\$a0	4	\$a0	3
	5044	\$ra	10040	\$t0	0
<b>sp</b> →	5040			\$v0	
	5036				
	5032				
	5028				
	5024				
	5020				
	5016				
	5012				
	5008				
	5004				
	5000				
	8000	jal fact			
	8004				
	...				
	10000	fact: addi \$sp, \$sp, -8	#adjust stack for 2 items		
<b>pc</b> →	10004	sw \$ra, 4(\$sp)	#save the return address		
	10008	sw \$a0, 0(\$sp)	#save the argument n		
	10012	slti \$t0,\$a0,1	#test for n < 1		
	10016	beq \$t0,\$zero,L1	#if n >= 1, go to L1		
	10020	addi \$v0,\$zero,1	#return 1		
	10024	addi \$sp,\$sp,8	#pop 2 items off stack		
	10028	jr \$ra	#return to caller		
	10032	L1: addi \$a0,\$a0,-1	#n >= 1: argument gets (n - 1)		
	10036	jal fact	#call fact with (n - 1)		
	10040	lw \$a0, 0(\$sp)	#return from jal: restore argument n		
	10044	lw \$ra, 4(\$sp)	#restore the return address		
	10048	addi \$sp, \$sp, 8	#adjust stack pointer to pop 2 items		
	10052	mul \$v0,\$a0,\$v0	#return n * fact (n - 1)		
	10056	jr \$ra	#return to the caller		

	5056				
	5052	\$ra	8004	\$ra	10040
	5048	\$a0	4	\$a0	3
	5044	\$ra	10040	\$t0	0
<b>sp→</b>	5040	\$a0	3	\$v0	
	5036				
	5032				
	5028				
	5024				
	5020				
	5016				
	5012				
	5008				
	5004				
	5000				
	8000	jal fact			
	8004				
	...				
	10000	fact: addi \$sp, \$sp, -8	#adjust stack for 2 items		
	10004	sw \$ra, 4(\$sp)	#save the return address		
<b>pc→</b>	10008	sw \$a0, 0(\$sp)	#save the argument n		
	10012	slti \$t0,\$a0,1	#test for n < 1		
	10016	beq \$t0,\$zero,L1	#if n >= 1, go to L1		
	10020	addi \$v0,\$zero,1	#return 1		
	10024	addi \$sp,\$sp,8	#pop 2 items off stack		
	10028	jr \$ra	#return to caller		
	10032	L1: addi \$a0,\$a0,-1	#n >= 1: argument gets (n - 1)		
	10036	jal fact	#call fact with (n - 1)		
	10040	lw \$a0, 0(\$sp)	#return from jal: restore argument n		
	10044	lw \$ra, 4(\$sp)	#restore the return address		
	10048	addi \$sp, \$sp, 8	#adjust stack pointer to pop 2 items		
	10052	mul \$v0,\$a0,\$v0	#return n * fact (n - 1)		
	10056	jr \$ra	#return to the caller		

	5056				
	5052	\$ra	8004	\$ra	10040
	5048	\$a0	4	\$a0	3
	5044	\$ra	10040	\$t0	0
<b>sp→</b>	5040	\$a0	3	\$v0	
	5036				
	5032				
	5028				
	5024				
	5020				
	5016				
	5012				
	5008				
	5004				
	5000				
	8000	jal fact			
	8004				
	...				
	10000	fact: addi \$sp, \$sp, -8	#adjust stack for 2 items		
	10004	sw \$ra, 4(\$sp)	#save the return address		
	10008	sw \$a0, 0(\$sp)	#save the argument n		
<b>pc→</b>	10012	slti \$t0,\$a0,1	#test for n < 1		
	10016	beq \$t0,\$zero,L1	#if n >= 1, go to L1		
	10020	addi \$v0,\$zero,1	#return 1		
	10024	addi \$sp,\$sp,8	#pop 2 items off stack		
	10028	jr \$ra	#return to caller		
	10032	L1: addi \$a0,\$a0,-1	#n >= 1: argument gets (n - 1)		
	10036	jal fact	#call fact with (n - 1)		
	10040	lw \$a0, 0(\$sp)	#return from jal: restore argument n		
	10044	lw \$ra, 4(\$sp)	#restore the return address		
	10048	addi \$sp, \$sp, 8	#adjust stack pointer to pop 2 items		
	10052	mul \$v0,\$a0,\$v0	#return n * fact (n - 1)		
	10056	jr \$ra	#return to the caller		

	5056				
	5052	\$ra	8004	\$ra	10040
	5048	\$a0	4	\$a0	3
	5044	\$ra	10040	\$t0	0
<b>sp→</b>	5040	\$a0	3	\$v0	
	5036				
	5032				
	5028				
	5024				
	5020				
	5016				
	5012				
	5008				
	5004				
	5000				
	8000	jal fact			
	8004				
	...				
	10000	fact: addi \$sp, \$sp, -8	#adjust stack for 2 items		
	10004	sw \$ra, 4(\$sp)	#save the return address		
	10008	sw \$a0, 0(\$sp)	#save the argument n		
	10012	slti \$t0,\$a0,1	#test for n < 1		
<b>pc→</b>	10016	beq \$t0,\$zero,L1	#if n >= 1, go to L1		
	10020	addi \$v0,\$zero,1	#return 1		
	10024	addi \$sp,\$sp,8	#pop 2 items off stack		
	10028	jr \$ra	#return to caller		
	10032	L1: addi \$a0,\$a0,-1	#n >= 1: argument gets (n - 1)		
	10036	jal fact	#call fact with (n - 1)		
	10040	lw \$a0, 0(\$sp)	#return from jal: restore argument n		
	10044	lw \$ra, 4(\$sp)	#restore the return address		
	10048	addi \$sp, \$sp, 8	#adjust stack pointer to pop 2 items		
	10052	mul \$v0,\$a0,\$v0	#return n * fact (n - 1)		
	10056	jr \$ra	#return to the caller		

	5056				
	5052	\$ra	8004	\$ra	10040
	5048	\$a0	4	\$a0	2
	5044	\$ra	10040	\$t0	0
<b>sp→</b>	5040	\$a0	3	\$v0	
	5036				
	5032				
	5028				
	5024				
	5020				
	5016				
	5012				
	5008				
	5004				
	5000				
	8000	jal fact			
	8004				
	...				
	10000	fact: addi \$sp, \$sp, -8	#adjust stack for 2 items		
	10004	sw \$ra, 4(\$sp)	#save the return address		
	10008	sw \$a0, 0(\$sp)	#save the argument n		
	10012	slti \$t0,\$a0,1	#test for n < 1		
	10016	beq \$t0,\$zero,L1	#if n >= 1, go to L1		
	10020	addi \$v0,\$zero,1	#return 1		
	10024	addi \$sp,\$sp,8	#pop 2 items off stack		
	10028	jr \$ra	#return to caller		
<b>pc→</b>	10032	L1: addi \$a0,\$a0,-1	#n >= 1: argument gets (n - 1)		
	10036	jal fact	#call fact with (n - 1)		
	10040	lw \$a0, 0(\$sp)	#return from jal: restore argument n		
	10044	lw \$ra, 4(\$sp)	#restore the return address		
	10048	addi \$sp, \$sp, 8	#adjust stack pointer to pop 2 items		
	10052	mul \$v0,\$a0,\$v0	#return n * fact (n - 1)		
	10056	jr \$ra	#return to the caller		

	5056				
	5052	\$ra	8004	\$ra	10040
	5048	\$a0	4	\$a0	2
	5044	\$ra	10040	\$t0	0
<b>sp→</b>	5040	\$a0	3	\$v0	
	5036				
	5032				
	5028				
	5024				
	5020				
	5016				
	5012				
	5008				
	5004				
	5000				
	8000	jal fact			
	8004				
	...				
	10000	fact: addi \$sp, \$sp, -8	#adjust stack for 2 items		
	10004	sw \$ra, 4(\$sp)	#save the return address		
	10008	sw \$a0, 0(\$sp)	#save the argument n		
	10012	slti \$t0,\$a0,1	#test for n < 1		
	10016	beq \$t0,\$zero,L1	#if n >= 1, go to L1		
	10020	addi \$v0,\$zero,1	#return 1		
	10024	addi \$sp,\$sp,8	#pop 2 items off stack		
	10028	jr \$ra	#return to caller		
	10032	L1: addi \$a0,\$a0,-1	#n >= 1: argument gets (n - 1)		
<b>pc→</b>	10036	jal fact	#call fact with (n - 1)		
	10040	lw \$a0, 0(\$sp)	#return from jal: restore argument n		
	10044	lw \$ra, 4(\$sp)	#restore the return address		
	10048	addi \$sp, \$sp, 8	#adjust stack pointer to pop 2 items		
	10052	mul \$v0,\$a0,\$v0	#return n * fact (n - 1)		
	10056	jr \$ra	#return to the caller		

	5056				
	5052	\$ra	8004	\$ra	10040
	5048	\$a0	4	\$a0	2
	5044	\$ra	10040	\$t0	0
	5040	\$a0	3	\$v0	
	5036				
<b>sp→</b>	5032				
	5028				
	5024				
	5020				
	5016				
	5012				
	5008				
	5004				
	5000				
	8000	jal fact			
	8004				
	...				
<b>pc→</b>	10000	fact: addi \$sp, \$sp, -8	#adjust stack for 2 items		
	10004	sw \$ra, 4(\$sp)	#save the return address		
	10008	sw \$a0, 0(\$sp)	#save the argument n		
	10012	slti \$t0,\$a0,1	#test for n < 1		
	10016	beq \$t0,\$zero,L1	#if n >= 1, go to L1		
	10020	addi \$v0,\$zero,1	#return 1		
	10024	addi \$sp,\$sp,8	#pop 2 items off stack		
	10028	jr \$ra	#return to caller		
	10032	L1: addi \$a0,\$a0,-1	#n >= 1: argument gets (n - 1)		
	10036	jal fact	#call fact with (n - 1)		
	10040	lw \$a0, 0(\$sp)	#return from jal: restore argument n		
	10044	lw \$ra, 4(\$sp)	#restore the return address		
	10048	addi \$sp, \$sp, 8	#adjust stack pointer to pop 2 items		
	10052	mul \$v0,\$a0,\$v0	#return n * fact (n - 1)		
	10056	jr \$ra	#return to the caller		



	5056				
	5052	\$ra	8004	\$ra	10040
	5048	\$a0	4	\$a0	2
	5044	\$ra	10040	\$t0	0
	5040	\$a0	3	\$v0	
	5036	\$ra	10040		
<b>sp→</b>	5032				
	5028				
	5024				
	5020				
	5016				
	5012				
	5008				
	5004				
	5000				
	8000	jal fact			
	8004				
	...				
	10000	fact: addi \$sp, \$sp, -8	#adjust stack for 2 items		
<b>pc→</b>	10004	sw \$ra, 4(\$sp)	#save the return address		
	10008	sw \$a0, 0(\$sp)	#save the argument n		
	10012	slti \$t0,\$a0,1	#test for n < 1		
	10016	beq \$t0,\$zero,L1	#if n >= 1, go to L1		
	10020	addi \$v0,\$zero,1	#return 1		
	10024	addi \$sp,\$sp,8	#pop 2 items off stack		
	10028	jr \$ra	#return to caller		
	10032	L1: addi \$a0,\$a0,-1	#n >= 1: argument gets (n - 1)		
	10036	jal fact	#call fact with (n - 1)		
	10040	lw \$a0, 0(\$sp)	#return from jal: restore argument n		
	10044	lw \$ra, 4(\$sp)	#restore the return address		
	10048	addi \$sp, \$sp, 8	#adjust stack pointer to pop 2 items		
	10052	mul \$v0,\$a0,\$v0	#return n * fact (n - 1)		
	10056	jr \$ra	#return to the caller		

	5056				
	5052	\$ra	8004	\$ra	10040
	5048	\$a0	4	\$a0	2
	5044	\$ra	10040	\$t0	0
	5040	\$a0	3	\$v0	
	5036	\$ra	10040		
<b>sp→</b>	5032	\$a0	2		
	5028				
	5024				
	5020				
	5016				
	5012				
	5008				
	5004				
	5000				
	8000	jal fact			
	8004				
	...				
	10000	fact: addi \$sp, \$sp, -8	#adjust stack for 2 items		
	10004	sw \$ra, 4(\$sp)	#save the return address		
<b>pc→</b>	10008	sw \$a0, 0(\$sp)	#save the argument n		
	10012	slti \$t0,\$a0,1	#test for n < 1		
	10016	beq \$t0,\$zero,L1	#if n >= 1, go to L1		
	10020	addi \$v0,\$zero,1	#return 1		
	10024	addi \$sp,\$sp,8	#pop 2 items off stack		
	10028	jr \$ra	#return to caller		
	10032	L1: addi \$a0,\$a0,-1	#n >= 1: argument gets (n - 1)		
	10036	jal fact	#call fact with (n - 1)		
	10040	lw \$a0, 0(\$sp)	#return from jal: restore argument n		
	10044	lw \$ra, 4(\$sp)	#restore the return address		
	10048	addi \$sp, \$sp, 8	#adjust stack pointer to pop 2 items		
	10052	mul \$v0,\$a0,\$v0	#return n * fact (n - 1)		
	10056	jr \$ra	#return to the caller		

	5056				
	5052	\$ra	8004	\$ra	10040
	5048	\$a0	4	\$a0	2
	5044	\$ra	10040	\$t0	0
	5040	\$a0	3	\$v0	
	5036	\$ra	10040		
<b>sp→</b>	5032	\$a0	2		
	5028				
	5024				
	5020				
	5016				
	5012				
	5008				
	5004				
	5000				
	8000	jal fact			
	8004				
	...				
	10000	fact: addi \$sp, \$sp, -8	#adjust stack for 2 items		
	10004	sw \$ra, 4(\$sp)	#save the return address		
	10008	sw \$a0, 0(\$sp)	#save the argument n		
<b>pc→</b>	10012	slti \$t0,\$a0,1	#test for n < 1		
	10016	beq \$t0,\$zero,L1	#if n >= 1, go to L1		
	10020	addi \$v0,\$zero,1	#return 1		
	10024	addi \$sp,\$sp,8	#pop 2 items off stack		
	10028	jr \$ra	#return to caller		
	10032	L1: addi \$a0,\$a0,-1	#n >= 1: argument gets (n - 1)		
	10036	jal fact	#call fact with (n - 1)		
	10040	lw \$a0, 0(\$sp)	#return from jal: restore argument n		
	10044	lw \$ra, 4(\$sp)	#restore the return address		
	10048	addi \$sp, \$sp, 8	#adjust stack pointer to pop 2 items		
	10052	mul \$v0,\$a0,\$v0	#return n * fact (n - 1)		
	10056	jr \$ra	#return to the caller		

	5056				
	5052	\$ra	8004	\$ra	10040
	5048	\$a0	4	\$a0	2
	5044	\$ra	10040	\$t0	0
	5040	\$a0	3	\$v0	
	5036	\$ra	10040		
<b>sp→</b>	5032	\$a0	2		
	5028				
	5024				
	5020				
	5016				
	5012				
	5008				
	5004				
	5000				
	8000	jal fact			
	8004				
	...				
	10000	fact: addi \$sp, \$sp, -8	#adjust stack for 2 items		
	10004	sw \$ra, 4(\$sp)	#save the return address		
	10008	sw \$a0, 0(\$sp)	#save the argument n		
	10012	slti \$t0,\$a0,1	#test for n < 1		
<b>pc→</b>	10016	beq \$t0,\$zero,L1	#if n >= 1, go to L1		
	10020	addi \$v0,\$zero,1	#return 1		
	10024	addi \$sp,\$sp,8	#pop 2 items off stack		
	10028	jr \$ra	#return to caller		
	10032	L1: addi \$a0,\$a0,-1	#n >= 1: argument gets (n - 1)		
	10036	jal fact	#call fact with (n - 1)		
	10040	lw \$a0, 0(\$sp)	#return from jal: restore argument n		
	10044	lw \$ra, 4(\$sp)	#restore the return address		
	10048	addi \$sp, \$sp, 8	#adjust stack pointer to pop 2 items		
	10052	mul \$v0,\$a0,\$v0	#return n * fact (n - 1)		
	10056	jr \$ra	#return to the caller		

	5056				
	5052	\$ra	8004	\$ra	10040
	5048	\$a0	4	\$a0	1
	5044	\$ra	10040	\$t0	0
	5040	\$a0	3	\$v0	
	5036	\$ra	10040		
<b>sp→</b>	5032	\$a0	2		
	5028				
	5024				
	5020				
	5016				
	5012				
	5008				
	5004				
	5000				
	8000	jal fact			
	8004				
	...				
	10000	fact: addi \$sp, \$sp, -8	#adjust stack for 2 items		
	10004	sw \$ra, 4(\$sp)	#save the return address		
	10008	sw \$a0, 0(\$sp)	#save the argument n		
	10012	slti \$t0,\$a0,1	#test for n < 1		
	10016	beq \$t0,\$zero,L1	#if n >= 1, go to L1		
	10020	addi \$v0,\$zero,1	#return 1		
	10024	addi \$sp,\$sp,8	#pop 2 items off stack		
	10028	jr \$ra	#return to caller		
<b>pc→</b>	10032	L1: addi \$a0,\$a0,-1	#n >= 1: argument gets (n - 1)		
	10036	jal fact	#call fact with (n - 1)		
	10040	lw \$a0, 0(\$sp)	#return from jal: restore argument n		
	10044	lw \$ra, 4(\$sp)	#restore the return address		
	10048	addi \$sp, \$sp, 8	#adjust stack pointer to pop 2 items		
	10052	mul \$v0,\$a0,\$v0	#return n * fact (n - 1)		
	10056	jr \$ra	#return to the caller		

	5056				
	5052	\$ra	8004	\$ra	10040
	5048	\$a0	4	\$a0	1
	5044	\$ra	10040	\$t0	0
	5040	\$a0	3	\$v0	
	5036	\$ra	10040		
<b>sp→</b>	5032	\$a0	2		
	5028				
	5024				
	5020				
	5016				
	5012				
	5008				
	5004				
	5000				
	8000	jal fact			
	8004				
	...				
	10000	fact: addi \$sp, \$sp, -8	#adjust stack for 2 items		
	10004	sw \$ra, 4(\$sp)	#save the return address		
	10008	sw \$a0, 0(\$sp)	#save the argument n		
	10012	slti \$t0,\$a0,1	#test for n < 1		
	10016	beq \$t0,\$zero,L1	#if n >= 1, go to L1		
	10020	addi \$v0,\$zero,1	#return 1		
	10024	addi \$sp,\$sp,8	#pop 2 items off stack		
	10028	jr \$ra	#return to caller		
	10032	L1: addi \$a0,\$a0,-1	#n >= 1: argument gets (n - 1)		
<b>pc→</b>	10036	jal fact	#call fact with (n - 1)		
	10040	lw \$a0, 0(\$sp)	#return from jal: restore argument n		
	10044	lw \$ra, 4(\$sp)	#restore the return address		
	10048	addi \$sp, \$sp, 8	#adjust stack pointer to pop 2 items		
	10052	mul \$v0,\$a0,\$v0	#return n * fact (n - 1)		
	10056	jr \$ra	#return to the caller		

	5056				
	5052	\$ra	8004	\$ra	10040
	5048	\$a0	4	\$a0	1
	5044	\$ra	10040	\$t0	0
	5040	\$a0	3	\$v0	
	5036	\$ra	10040		
	5032	\$a0	2		
	5028				
<b>sp→</b>	5024				
	5020				
	5016				
	5012				
	5008				
	5004				
	5000				
	8000	jal fact			
	8004				
	...				
<b>pc→</b>	10000	fact: addi \$sp, \$sp, -8	#adjust stack for 2 items		
	10004	sw \$ra, 4(\$sp)	#save the return address		
	10008	sw \$a0, 0(\$sp)	#save the argument n		
	10012	slti \$t0,\$a0,1	#test for n < 1		
	10016	beq \$t0,\$zero,L1	#if n >= 1, go to L1		
	10020	addi \$v0,\$zero,1	#return 1		
	10024	addi \$sp,\$sp,8	#pop 2 items off stack		
	10028	jr \$ra	#return to caller		
	10032	L1: addi \$a0,\$a0,-1	#n >= 1: argument gets (n - 1)		
	10036	jal fact	#call fact with (n - 1)		
	10040	lw \$a0, 0(\$sp)	#return from jal: restore argument n		
	10044	lw \$ra, 4(\$sp)	#restore the return address		
	10048	addi \$sp, \$sp, 8	#adjust stack pointer to pop 2 items		
	10052	mul \$v0,\$a0,\$v0	#return n * fact (n - 1)		
	10056	jr \$ra	#return to the caller		

	5056				
	5052	\$ra	8004	\$ra	10040
	5048	\$a0	4	\$a0	1
	5044	\$ra	10040	\$t0	0
	5040	\$a0	3	\$v0	
	5036	\$ra	10040		
	5032	\$a0	2		
	5028	\$ra	10040		
<b>sp→</b>	5024				
	5020				
	5016				
	5012				
	5008				
	5004				
	5000				
	8000	jal fact			
	8004				
	...				
	10000	fact: addi \$sp, \$sp, -8	#adjust stack for 2 items		
<b>pc→</b>	10004	sw \$ra, 4(\$sp)	#save the return address		
	10008	sw \$a0, 0(\$sp)	#save the argument n		
	10012	slti \$t0,\$a0,1	#test for n < 1		
	10016	beq \$t0,\$zero,L1	#if n >= 1, go to L1		
	10020	addi \$v0,\$zero,1	#return 1		
	10024	addi \$sp,\$sp,8	#pop 2 items off stack		
	10028	jr \$ra	#return to caller		
	10032	L1: addi \$a0,\$a0,-1	#n >= 1: argument gets (n - 1)		
	10036	jal fact	#call fact with (n - 1)		
	10040	lw \$a0, 0(\$sp)	#return from jal: restore argument n		
	10044	lw \$ra, 4(\$sp)	#restore the return address		
	10048	addi \$sp, \$sp, 8	#adjust stack pointer to pop 2 items		
	10052	mul \$v0,\$a0,\$v0	#return n * fact (n - 1)		
	10056	jr \$ra	#return to the caller		



	5056				
	5052	\$ra	8004	\$ra	10040
	5048	\$a0	4	\$a0	1
	5044	\$ra	10040	\$t0	0
	5040	\$a0	3	\$v0	
	5036	\$ra	10040		
	5032	\$a0	2		
	5028	\$ra	10040		
<b>sp→</b>	5024	\$a0	1		
	5020				
	5016				
	5012				
	5008				
	5004				
	5000				
	8000	jal fact			
	8004				
	...				
	10000	fact: addi \$sp, \$sp, -8	#adjust stack for 2 items		
	10004	sw \$ra, 4(\$sp)	#save the return address		
<b>pc→</b>	10008	sw \$a0, 0(\$sp)	#save the argument n		
	10012	slti \$t0,\$a0,1	#test for n < 1		
	10016	beq \$t0,\$zero,L1	#if n >= 1, go to L1		
	10020	addi \$v0,\$zero,1	#return 1		
	10024	addi \$sp,\$sp,8	#pop 2 items off stack		
	10028	jr \$ra	#return to caller		
	10032	L1: addi \$a0,\$a0,-1	#n >= 1: argument gets (n - 1)		
	10036	jal fact	#call fact with (n - 1)		
	10040	lw \$a0, 0(\$sp)	#return from jal: restore argument n		
	10044	lw \$ra, 4(\$sp)	#restore the return address		
	10048	addi \$sp, \$sp, 8	#adjust stack pointer to pop 2 items		
	10052	mul \$v0,\$a0,\$v0	#return n * fact (n - 1)		
	10056	jr \$ra	#return to the caller		

	5056				
	5052	\$ra	8004	\$ra	10040
	5048	\$a0	4	\$a0	1
	5044	\$ra	10040	\$t0	0
	5040	\$a0	3	\$v0	
	5036	\$ra	10040		
	5032	\$a0	2		
	5028	\$ra	10040		
<b>sp→</b>	5024	\$a0	1		
	5020				
	5016				
	5012				
	5008				
	5004				
	5000				
	8000	jal fact			
	8004				
	...				
	10000	fact: addi \$sp, \$sp, -8	#adjust stack for 2 items		
	10004	sw \$ra, 4(\$sp)	#save the return address		
	10008	sw \$a0, 0(\$sp)	#save the argument n		
<b>pc→</b>	10012	slti \$t0,\$a0,1	#test for n < 1		
	10016	beq \$t0,\$zero,L1	#if n >= 1, go to L1		
	10020	addi \$v0,\$zero,1	#return 1		
	10024	addi \$sp,\$sp,8	#pop 2 items off stack		
	10028	jr \$ra	#return to caller		
	10032	L1: addi \$a0,\$a0,-1	#n >= 1: argument gets (n - 1)		
	10036	jal fact	#call fact with (n - 1)		
	10040	lw \$a0, 0(\$sp)	#return from jal: restore argument n		
	10044	lw \$ra, 4(\$sp)	#restore the return address		
	10048	addi \$sp, \$sp, 8	#adjust stack pointer to pop 2 items		
	10052	mul \$v0,\$a0,\$v0	#return n * fact (n - 1)		
	10056	jr \$ra	#return to the caller		

	5056				
	5052	\$ra	8004	\$ra	10040
	5048	\$a0	4	\$a0	1
	5044	\$ra	10040	\$t0	0
	5040	\$a0	3	\$v0	
	5036	\$ra	10040		
	5032	\$a0	2		
	5028	\$ra	10040		
<b>sp→</b>	5024	\$a0	1		
	5020				
	5016				
	5012				
	5008				
	5004				
	5000				
	8000	jal fact			
	8004				
	...				
	10000	fact: addi \$sp, \$sp, -8	#adjust stack for 2 items		
	10004	sw \$ra, 4(\$sp)	#save the return address		
	10008	sw \$a0, 0(\$sp)	#save the argument n		
	10012	slti \$t0,\$a0,1	#test for n < 1		
<b>pc→</b>	10016	beq \$t0,\$zero,L1	#if n >= 1, go to L1		
	10020	addi \$v0,\$zero,1	#return 1		
	10024	addi \$sp,\$sp,8	#pop 2 items off stack		
	10028	jr \$ra	#return to caller		
	10032	L1: addi \$a0,\$a0,-1	#n >= 1: argument gets (n - 1)		
	10036	jal fact	#call fact with (n - 1)		
	10040	lw \$a0, 0(\$sp)	#return from jal: restore argument n		
	10044	lw \$ra, 4(\$sp)	#restore the return address		
	10048	addi \$sp, \$sp, 8	#adjust stack pointer to pop 2 items		
	10052	mul \$v0,\$a0,\$v0	#return n * fact (n - 1)		
	10056	jr \$ra	#return to the caller		

	5056				
	5052	\$ra	8004	\$ra	10040
	5048	\$a0	4	\$a0	0
	5044	\$ra	10040	\$t0	0
	5040	\$a0	3	\$v0	
	5036	\$ra	10040		
	5032	\$a0	2		
	5028	\$ra	10040		
<b>sp→</b>	5024	\$a0	1		
	5020				
	5016				
	5012				
	5008				
	5004				
	5000				
	8000	jal fact			
	8004				
	...				
	10000	fact: addi \$sp, \$sp, -8	#adjust stack for 2 items		
	10004	sw \$ra, 4(\$sp)	#save the return address		
	10008	sw \$a0, 0(\$sp)	#save the argument n		
	10012	slti \$t0,\$a0,1	#test for n < 1		
	10016	beq \$t0,\$zero,L1	#if n >= 1, go to L1		
	10020	addi \$v0,\$zero,1	#return 1		
	10024	addi \$sp,\$sp,8	#pop 2 items off stack		
	10028	jr \$ra	#return to caller		
<b>pc→</b>	10032	L1: addi \$a0,\$a0,-1	#n >= 1: argument gets (n - 1)		
	10036	jal fact	#call fact with (n - 1)		
	10040	lw \$a0, 0(\$sp)	#return from jal: restore argument n		
	10044	lw \$ra, 4(\$sp)	#restore the return address		
	10048	addi \$sp, \$sp, 8	#adjust stack pointer to pop 2 items		
	10052	mul \$v0,\$a0,\$v0	#return n * fact (n - 1)		
	10056	jr \$ra	#return to the caller		

	5056				
	5052	\$ra	8004	\$ra	10040
	5048	\$a0	4	\$a0	0
	5044	\$ra	10040	\$t0	0
	5040	\$a0	3	\$v0	
	5036	\$ra	10040		
	5032	\$a0	2		
	5028	\$ra	10040		
<b>sp→</b>	5024	\$a0	1		
	5020				
	5016				
	5012				
	5008				
	5004				
	5000				
	8000	jal fact			
	8004				
	...				
	10000	fact: addi \$sp, \$sp, -8	#adjust stack for 2 items		
	10004	sw \$ra, 4(\$sp)	#save the return address		
	10008	sw \$a0, 0(\$sp)	#save the argument n		
	10012	slti \$t0,\$a0,1	#test for n < 1		
	10016	beq \$t0,\$zero,L1	#if n >= 1, go to L1		
	10020	addi \$v0,\$zero,1	#return 1		
	10024	addi \$sp,\$sp,8	#pop 2 items off stack		
	10028	jr \$ra	#return to caller		
	10032	L1: addi \$a0,\$a0,-1	#n >= 1: argument gets (n - 1)		
<b>pc→</b>	10036	jal fact	#call fact with (n - 1)		
	10040	lw \$a0, 0(\$sp)	#return from jal: restore argument n		
	10044	lw \$ra, 4(\$sp)	#restore the return address		
	10048	addi \$sp, \$sp, 8	#adjust stack pointer to pop 2 items		
	10052	mul \$v0,\$a0,\$v0	#return n * fact (n - 1)		
	10056	jr \$ra	#return to the caller		

	5056				
	5052	\$ra	8004	\$ra	10040
	5048	\$a0	4	\$a0	0
	5044	\$ra	10040	\$t0	0
	5040	\$a0	3	\$v0	
	5036	\$ra	10040		
	5032	\$a0	2		
	5028	\$ra	10040		
	5024	\$a0	1		
	5020				
<b>sp→</b>	5016				
	5012				
	5008				
	5004				
	5000				
	8000	jal fact			
	8004				
	...				
<b>pc→</b>	10000	fact: addi \$sp, \$sp, -8	#adjust stack for 2 items		
	10004	sw \$ra, 4(\$sp)	#save the return address		
	10008	sw \$a0, 0(\$sp)	#save the argument n		
	10012	slti \$t0,\$a0,1	#test for n < 1		
	10016	beq \$t0,\$zero,L1	#if n >= 1, go to L1		
	10020	addi \$v0,\$zero,1	#return 1		
	10024	addi \$sp,\$sp,8	#pop 2 items off stack		
	10028	jr \$ra	#return to caller		
	10032	L1: addi \$a0,\$a0,-1	#n >= 1: argument gets (n - 1)		
	10036	jal fact	#call fact with (n - 1)		
	10040	lw \$a0, 0(\$sp)	#return from jal: restore argument n		
	10044	lw \$ra, 4(\$sp)	#restore the return address		
	10048	addi \$sp, \$sp, 8	#adjust stack pointer to pop 2 items		
	10052	mul \$v0,\$a0,\$v0	#return n * fact (n - 1)		
	10056	jr \$ra	#return to the caller		

	5056				
	5052	\$ra	8004	\$ra	10040
	5048	\$a0	4	\$a0	0
	5044	\$ra	10040	\$t0	0
	5040	\$a0	3	\$v0	
	5036	\$ra	10040		
	5032	\$a0	2		
	5028	\$ra	10040		
	5024	\$a0	1		
	5020	\$ra	10040		
<b>sp→</b>	5016				
	5012				
	5008				
	5004				
	5000				
	8000	jal fact			
	8004				
	...				
	10000	fact: addi \$sp, \$sp, -8	#adjust stack for 2 items		
<b>pc→</b>	10004	sw \$ra, 4(\$sp)	#save the return address		
	10008	sw \$a0, 0(\$sp)	#save the argument n		
	10012	slti \$t0,\$a0,1	#test for n < 1		
	10016	beq \$t0,\$zero,L1	#if n >= 1, go to L1		
	10020	addi \$v0,\$zero,1	#return 1		
	10024	addi \$sp,\$sp,8	#pop 2 items off stack		
	10028	jr \$ra	#return to caller		
	10032	L1: addi \$a0,\$a0,-1	#n >= 1: argument gets (n - 1)		
	10036	jal fact	#call fact with (n - 1)		
	10040	lw \$a0, 0(\$sp)	#return from jal: restore argument n		
	10044	lw \$ra, 4(\$sp)	#restore the return address		
	10048	addi \$sp, \$sp, 8	#adjust stack pointer to pop 2 items		
	10052	mul \$v0,\$a0,\$v0	#return n * fact (n - 1)		
	10056	jr \$ra	#return to the caller		

	5056				
	5052	\$ra	8004	\$ra	10040
	5048	\$a0	4	\$a0	0
	5044	\$ra	10040	\$t0	0
	5040	\$a0	3	\$v0	
	5036	\$ra	10040		
	5032	\$a0	2		
	5028	\$ra	10040		
	5024	\$a0	1		
	5020	\$ra	10040		
<b>sp→</b>	5016	\$a0	0		
	5012				
	5008				
	5004				
	5000				
	8000	jal fact			
	8004				
	...				
	10000	fact: addi \$sp, \$sp, -8	#adjust stack for 2 items		
	10004	sw \$ra, 4(\$sp)	#save the return address		
<b>pc→</b>	10008	sw \$a0, 0(\$sp)	#save the argument n		
	10012	slti \$t0,\$a0,1	#test for n < 1		
	10016	beq \$t0,\$zero,L1	#if n >= 1, go to L1		
	10020	addi \$v0,\$zero,1	#return 1		
	10024	addi \$sp,\$sp,8	#pop 2 items off stack		
	10028	jr \$ra	#return to caller		
	10032	L1: addi \$a0,\$a0,-1	#n >= 1: argument gets (n - 1)		
	10036	jal fact	#call fact with (n - 1)		
	10040	lw \$a0, 0(\$sp)	#return from jal: restore argument n		
	10044	lw \$ra, 4(\$sp)	#restore the return address		
	10048	addi \$sp, \$sp, 8	#adjust stack pointer to pop 2 items		
	10052	mul \$v0,\$a0,\$v0	#return n * fact (n - 1)		
	10056	jr \$ra	#return to the caller		



	5056				
	5052	\$ra	8004	\$ra	10040
	5048	\$a0	4	\$a0	0
	5044	\$ra	10040	\$t0	1
	5040	\$a0	3	\$v0	
	5036	\$ra	10040		
	5032	\$a0	2		
	5028	\$ra	10040		
	5024	\$a0	1		
	5020	\$ra	10040		
<b>sp→</b>	5016	\$a0	0		
	5012				
	5008				
	5004				
	5000				
	8000	jal fact			
	8004				
	...				
	10000	fact: addi \$sp, \$sp, -8	#adjust stack for 2 items		
	10004	sw \$ra, 4(\$sp)	#save the return address		
	10008	sw \$a0, 0(\$sp)	#save the argument n		
<b>pc→</b>	10012	slti \$t0,\$a0,1	#test for n < 1		
	10016	beq \$t0,\$zero,L1	#if n >= 1, go to L1		
	10020	addi \$v0,\$zero,1	#return 1		
	10024	addi \$sp,\$sp,8	#pop 2 items off stack		
	10028	jr \$ra	#return to caller		
	10032	L1: addi \$a0,\$a0,-1	#n >= 1: argument gets (n - 1)		
	10036	jal fact	#call fact with (n - 1)		
	10040	lw \$a0, 0(\$sp)	#return from jal: restore argument n		
	10044	lw \$ra, 4(\$sp)	#restore the return address		
	10048	addi \$sp, \$sp, 8	#adjust stack pointer to pop 2 items		
	10052	mul \$v0,\$a0,\$v0	#return n * fact (n - 1)		
	10056	jr \$ra	#return to the caller		

	5056				
	5052	\$ra	8004	\$ra	10040
	5048	\$a0	4	\$a0	0
	5044	\$ra	10040	\$t0	1
	5040	\$a0	3	\$v0	
	5036	\$ra	10040		
	5032	\$a0	2		
	5028	\$ra	10040		
	5024	\$a0	1		
	5020	\$ra	10040		
<b>sp→</b>	5016	\$a0	0		
	5012				
	5008				
	5004				
	5000				
	8000	jal fact			
	8004				
	...				
	10000	fact: addi \$sp, \$sp, -8	#adjust stack for 2 items		
	10004	sw \$ra, 4(\$sp)	#save the return address		
	10008	sw \$a0, 0(\$sp)	#save the argument n		
	10012	slti \$t0,\$a0,1	#test for n < 1		
<b>pc→</b>	10016	beq \$t0,\$zero,L1	#if n >= 1, go to L1		
	10020	addi \$v0,\$zero,1	#return 1		
	10024	addi \$sp,\$sp,8	#pop 2 items off stack		
	10028	jr \$ra	#return to caller		
	10032	L1: addi \$a0,\$a0,-1	#n >= 1: argument gets (n - 1)		
	10036	jal fact	#call fact with (n - 1)		
	10040	lw \$a0, 0(\$sp)	#return from jal: restore argument n		
	10044	lw \$ra, 4(\$sp)	#restore the return address		
	10048	addi \$sp, \$sp, 8	#adjust stack pointer to pop 2 items		
	10052	mul \$v0,\$a0,\$v0	#return n * fact (n - 1)		
	10056	jr \$ra	#return to the caller		

	5056				
	5052	\$ra	8004	\$ra	10040
	5048	\$a0	4	\$a0	0
	5044	\$ra	10040	\$t0	1
	5040	\$a0	3	\$v0	1
	5036	\$ra	10040		
	5032	\$a0	2		
	5028	\$ra	10040		
	5024	\$a0	1		
	5020	\$ra	10040		
<b>sp→</b>	5016	\$a0	0		
	5012				
	5008				
	5004				
	5000				
	8000	jal fact			
	8004				
	...				
	10000	fact: addi \$sp, \$sp, -8	#adjust stack for 2 items		
	10004	sw \$ra, 4(\$sp)	#save the return address		
	10008	sw \$a0, 0(\$sp)	#save the argument n		
	10012	slti \$t0,\$a0,1	#test for n < 1		
	10016	beq \$t0,\$zero,L1	#if n >= 1, go to L1		
<b>pc→</b>	10020	addi \$v0,\$zero,1	#return 1		
	10024	addi \$sp,\$sp,8	#pop 2 items off stack		
	10028	jr \$ra	#return to caller		
	10032	L1: addi \$a0,\$a0,-1	#n >= 1: argument gets (n - 1)		
	10036	jal fact	#call fact with (n - 1)		
	10040	lw \$a0, 0(\$sp)	#return from jal: restore argument n		
	10044	lw \$ra, 4(\$sp)	#restore the return address		
	10048	addi \$sp, \$sp, 8	#adjust stack pointer to pop 2 items		
	10052	mul \$v0,\$a0,\$v0	#return n * fact (n - 1)		
	10056	jr \$ra	#return to the caller		

	5056				
	5052	\$ra	8004	\$ra	10040
	5048	\$a0	4	\$a0	0
	5044	\$ra	10040	\$t0	1
	5040	\$a0	3	\$v0	1
	5036	\$ra	10040		
	5032	\$a0	2		
	5028	\$ra	10040		
<b>sp→</b>	5024	\$a0	1		
	5020	\$ra	10040		
	5016	\$a0	0		
	5012				
	5008				
	5004				
	5000				
	8000	jal fact			
	8004				
	...				
	10000	fact: addi \$sp, \$sp, -8	#adjust stack for 2 items		
	10004	sw \$ra, 4(\$sp)	#save the return address		
	10008	sw \$a0, 0(\$sp)	#save the argument n		
	10012	slti \$t0,\$a0,1	#test for n < 1		
	10016	beq \$t0,\$zero,L1	#if n >= 1, go to L1		
	10020	addi \$v0,\$zero,1	#return 1		
<b>pc→</b>	10024	addi \$sp,\$sp,8	#pop 2 items off stack		
	10028	jr \$ra	#return to caller		
	10032	L1: addi \$a0,\$a0,-1	#n >= 1: argument gets (n - 1)		
	10036	jal fact	#call fact with (n - 1)		
	10040	lw \$a0, 0(\$sp)	#return from jal: restore argument n		
	10044	lw \$ra, 4(\$sp)	#restore the return address		
	10048	addi \$sp, \$sp, 8	#adjust stack pointer to pop 2 items		
	10052	mul \$v0,\$a0,\$v0	#return n * fact (n - 1)		
	10056	jr \$ra	#return to the caller		

	5056				
	5052	\$ra	8004	\$ra	10040
	5048	\$a0	4	\$a0	0
	5044	\$ra	10040	\$t0	1
	5040	\$a0	3	\$v0	1
	5036	\$ra	10040		
	5032	\$a0	2		
	5028	\$ra	10040		
<b>sp→</b>	5024	\$a0	1		
	5020	\$ra	10040		
	5016	\$a0	0		
	5012				
	5008				
	5004				
	5000				
	8000	jal fact			
	8004				
	...				
	10000	fact: addi \$sp, \$sp, -8	#adjust stack for 2 items		
	10004	sw \$ra, 4(\$sp)	#save the return address		
	10008	sw \$a0, 0(\$sp)	#save the argument n		
	10012	slti \$t0,\$a0,1	#test for n < 1		
	10016	beq \$t0,\$zero,L1	#if n >= 1, go to L1		
	10020	addi \$v0,\$zero,1	#return 1		
	10024	addi \$sp,\$sp,8	#pop 2 items off stack		
<b>pc→</b>	10028	jr \$ra	#return to caller		
	10032	L1: addi \$a0,\$a0,-1	#n >= 1: argument gets (n - 1)		
	10036	jal fact	#call fact with (n - 1)		
	10040	lw \$a0, 0(\$sp)	#return from jal: restore argument n		
	10044	lw \$ra, 4(\$sp)	#restore the return address		
	10048	addi \$sp, \$sp, 8	#adjust stack pointer to pop 2 items		
	10052	mul \$v0,\$a0,\$v0	#return n * fact (n - 1)		
	10056	jr \$ra	#return to the caller		

	5056				
	5052	\$ra	8004	\$ra	10040
	5048	\$a0	4	\$a0	1
	5044	\$ra	10040	\$t0	1
	5040	\$a0	3	\$v0	1
	5036	\$ra	10040		
	5032	\$a0	2		
	5028	\$ra	10040		
sp→	5024	\$a0	1		
	5020	\$ra	10040		
	5016	\$a0	0		
	5012				
	5008				
	5004				
	5000				
	8000	jal fact			
	8004				
	...				
	10000	fact: addi \$sp, \$sp, -8	#adjust stack for 2 items		
	10004	sw \$ra, 4(\$sp)	#save the return address		
	10008	sw \$a0, 0(\$sp)	#save the argument n		
	10012	slti \$t0,\$a0,1	#test for n < 1		
	10016	beq \$t0,\$zero,L1	#if n >= 1, go to L1		
	10020	addi \$v0,\$zero,1	#return 1		
	10024	addi \$sp,\$sp,8	#pop 2 items off stack		
	10028	jr \$ra	#return to caller		
	10032	L1: addi \$a0,\$a0,-1	#n >= 1: argument gets (n - 1)		
	10036	jal fact	#call fact with (n - 1)		
pc→	10040	lw \$a0, 0(\$sp)	#return from jal: restore argument n		
	10044	lw \$ra, 4(\$sp)	#restore the return address		
	10048	addi \$sp, \$sp, 8	#adjust stack pointer to pop 2 items		
	10052	mul \$v0,\$a0,\$v0	#return n * fact (n - 1)		
	10056	jr \$ra	#return to the caller		

	5056				
	5052	\$ra	8004	\$ra	10040
	5048	\$a0	4	\$a0	1
	5044	\$ra	10040	\$t0	1
	5040	\$a0	3	\$v0	1
	5036	\$ra	10040		
	5032	\$a0	2		
	5028	\$ra	10040		
<b>sp→</b>	5024	\$a0	1		
	5020	\$ra	10040		
	5016	\$a0	0		
	5012				
	5008				
	5004				
	5000				
	8000	jal fact			
	8004				
	...				
	10000	fact: addi \$sp, \$sp, -8	#adjust stack for 2 items		
	10004	sw \$ra, 4(\$sp)	#save the return address		
	10008	sw \$a0, 0(\$sp)	#save the argument n		
	10012	slti \$t0,\$a0,1	#test for n < 1		
	10016	beq \$t0,\$zero,L1	#if n >= 1, go to L1		
	10020	addi \$v0,\$zero,1	#return 1		
	10024	addi \$sp,\$sp,8	#pop 2 items off stack		
	10028	jr \$ra	#return to caller		
	10032	L1: addi \$a0,\$a0,-1	#n >= 1: argument gets (n - 1)		
	10036	jal fact	#call fact with (n - 1)		
	10040	lw \$a0, 0(\$sp)	#return from jal: restore argument n		
<b>pc→</b>	10044	lw \$ra, 4(\$sp)	#restore the return address		
	10048	addi \$sp, \$sp, 8	#adjust stack pointer to pop 2 items		
	10052	mul \$v0,\$a0,\$v0	#return n * fact (n - 1)		
	10056	jr \$ra	#return to the caller		

	5056				
	5052	\$ra	8004	\$ra	10040
	5048	\$a0	4	\$a0	1
	5044	\$ra	10040	\$t0	1
	5040	\$a0	3	\$v0	1
	5036	\$ra	10040		
<b>sp→</b>	5032	\$a0	2		
	5028	\$ra	10040		
	5024	\$a0	1		
	5020	\$ra	10040		
	5016	\$a0	0		
	5012				
	5008				
	5004				
	5000				
	8000	jal fact			
	8004				
	...				
	10000	fact: addi \$sp, \$sp, -8	#adjust stack for 2 items		
	10004	sw \$ra, 4(\$sp)	#save the return address		
	10008	sw \$a0, 0(\$sp)	#save the argument n		
	10012	slti \$t0,\$a0,1	#test for n < 1		
	10016	beq \$t0,\$zero,L1	#if n >= 1, go to L1		
	10020	addi \$v0,\$zero,1	#return 1		
	10024	addi \$sp,\$sp,8	#pop 2 items off stack		
	10028	jr \$ra	#return to caller		
	10032	L1: addi \$a0,\$a0,-1	#n >= 1: argument gets (n - 1)		
	10036	jal fact	#call fact with (n - 1)		
	10040	lw \$a0, 0(\$sp)	#return from jal: restore argument n		
	10044	lw \$ra, 4(\$sp)	#restore the return address		
<b>pc→</b>	10048	addi \$sp, \$sp, 8	#adjust stack pointer to pop 2 items		
	10052	mul \$v0,\$a0,\$v0	#return n * fact (n - 1)		
	10056	jr \$ra	#return to the caller		



	5056				
	5052	\$ra	8004	\$ra	10040
	5048	\$a0	4	\$a0	1
	5044	\$ra	10040	\$t0	1
	5040	\$a0	3	\$v0	1
	5036	\$ra	10040		
<b>sp→</b>	5032	\$a0	2		
	5028	\$ra	10040		
	5024	\$a0	1		
	5020	\$ra	10040		
	5016	\$a0	0		
	5012				
	5008				
	5004				
	5000				
	8000	jal fact			
	8004				
	...				
	10000	fact: addi \$sp, \$sp, -8	#adjust stack for 2 items		
	10004	sw \$ra, 4(\$sp)	#save the return address		
	10008	sw \$a0, 0(\$sp)	#save the argument n		
	10012	slti \$t0,\$a0,1	#test for n < 1		
	10016	beq \$t0,\$zero,L1	#if n >= 1, go to L1		
	10020	addi \$v0,\$zero,1	#return 1		
	10024	addi \$sp,\$sp,8	#pop 2 items off stack		
	10028	jr \$ra	#return to caller		
	10032	L1: addi \$a0,\$a0,-1	#n >= 1: argument gets (n - 1)		
	10036	jal fact	#call fact with (n - 1)		
	10040	lw \$a0, 0(\$sp)	#return from jal: restore argument n		
	10044	lw \$ra, 4(\$sp)	#restore the return address		
	10048	addi \$sp, \$sp, 8	#adjust stack pointer to pop 2 items		
<b>pc→</b>	10052	mul \$v0,\$a0,\$v0	#return n * fact (n - 1)		
	10056	jr \$ra	#return to the caller		

	5056				
	5052	\$ra	8004	\$ra	10040
	5048	\$a0	4	\$a0	1
	5044	\$ra	10040	\$t0	1
	5040	\$a0	3	\$v0	1
	5036	\$ra	10040		
<b>sp→</b>	5032	\$a0	2		
	5028	\$ra	10040		
	5024	\$a0	1		
	5020	\$ra	10040		
	5016	\$a0	0		
	5012				
	5008				
	5004				
	5000				
	8000	jal fact			
	8004				
	...				
	10000	fact: addi \$sp, \$sp, -8	#adjust stack for 2 items		
	10004	sw \$ra, 4(\$sp)	#save the return address		
	10008	sw \$a0, 0(\$sp)	#save the argument n		
	10012	slti \$t0,\$a0,1	#test for n < 1		
	10016	beq \$t0,\$zero,L1	#if n >= 1, go to L1		
	10020	addi \$v0,\$zero,1	#return 1		
	10024	addi \$sp,\$sp,8	#pop 2 items off stack		
	10028	jr \$ra	#return to caller		
	10032	L1: addi \$a0,\$a0,-1	#n >= 1: argument gets (n - 1)		
	10036	jal fact	#call fact with (n - 1)		
	10040	lw \$a0, 0(\$sp)	#return from jal: restore argument n		
	10044	lw \$ra, 4(\$sp)	#restore the return address		
	10048	addi \$sp, \$sp, 8	#adjust stack pointer to pop 2 items		
	10052	mul \$v0,\$a0,\$v0	#return n * fact (n - 1)		
<b>pc→</b>	10056	jr \$ra	#return to the caller		

	5056				
	5052	\$ra	8004	\$ra	10040
	5048	\$a0	4	\$a0	2
	5044	\$ra	10040	\$t0	1
	5040	\$a0	3	\$v0	1
	5036	\$ra	10040		
<b>sp→</b>	5032	\$a0	2		
	5028	\$ra	10040		
	5024	\$a0	1		
	5020	\$ra	10040		
	5016	\$a0	0		
	5012				
	5008				
	5004				
	5000				
	8000	jal fact			
	8004				
	...				
	10000	fact: addi \$sp, \$sp, -8	#adjust stack for 2 items		
	10004	sw \$ra, 4(\$sp)	#save the return address		
	10008	sw \$a0, 0(\$sp)	#save the argument n		
	10012	slti \$t0,\$a0,1	#test for n < 1		
	10016	beq \$t0,\$zero,L1	#if n >= 1, go to L1		
	10020	addi \$v0,\$zero,1	#return 1		
	10024	addi \$sp,\$sp,8	#pop 2 items off stack		
	10028	jr \$ra	#return to caller		
	10032	L1: addi \$a0,\$a0,-1	#n >= 1: argument gets (n - 1)		
	10036	jal fact	#call fact with (n - 1)		
<b>pc→</b>	10040	lw \$a0, 0(\$sp)	#return from jal: restore argument n		
	10044	lw \$ra, 4(\$sp)	#restore the return address		
	10048	addi \$sp, \$sp, 8	#adjust stack pointer to pop 2 items		
	10052	mul \$v0,\$a0,\$v0	#return n * fact (n - 1)		
	10056	jr \$ra	#return to the caller		

	5056				
	5052	\$ra	8004	\$ra	10040
	5048	\$a0	4	\$a0	2
	5044	\$ra	10040	\$t0	1
	5040	\$a0	3	\$v0	1
	5036	\$ra	10040		
<b>sp→</b>	5032	\$a0	2		
	5028	\$ra	10040		
	5024	\$a0	1		
	5020	\$ra	10040		
	5016	\$a0	0		
	5012				
	5008				
	5004				
	5000				
	8000	jal fact			
	8004				
	...				
	10000	fact: addi \$sp, \$sp, -8	#adjust stack for 2 items		
	10004	sw \$ra, 4(\$sp)	#save the return address		
	10008	sw \$a0, 0(\$sp)	#save the argument n		
	10012	slti \$t0,\$a0,1	#test for n < 1		
	10016	beq \$t0,\$zero,L1	#if n >= 1, go to L1		
	10020	addi \$v0,\$zero,1	#return 1		
	10024	addi \$sp,\$sp,8	#pop 2 items off stack		
	10028	jr \$ra	#return to caller		
	10032	L1: addi \$a0,\$a0,-1	#n >= 1: argument gets (n - 1)		
	10036	jal fact	#call fact with (n - 1)		
	10040	lw \$a0, 0(\$sp)	#return from jal: restore argument n		
<b>pc→</b>	10044	lw \$ra, 4(\$sp)	#restore the return address		
	10048	addi \$sp, \$sp, 8	#adjust stack pointer to pop 2 items		
	10052	mul \$v0,\$a0,\$v0	#return n * fact (n - 1)		
	10056	jr \$ra	#return to the caller		

	5056				
	5052	\$ra	8004	\$ra	10040
	5048	\$a0	4	\$a0	2
	5044	\$ra	10040	\$t0	1
<b>sp→</b>	5040	\$a0	3	\$v0	1
	5036	\$ra	10040		
	5032	\$a0	2		
	5028	\$ra	10040		
	5024	\$a0	1		
	5020	\$ra	10040		
	5016	\$a0	0		
	5012				
	5008				
	5004				
	5000				
	8000	jal fact			
	8004				
	...				
	10000	fact: addi \$sp, \$sp, -8	#adjust stack for 2 items		
	10004	sw \$ra, 4(\$sp)	#save the return address		
	10008	sw \$a0, 0(\$sp)	#save the argument n		
	10012	slti \$t0,\$a0,1	#test for n < 1		
	10016	beq \$t0,\$zero,L1	#if n >= 1, go to L1		
	10020	addi \$v0,\$zero,1	#return 1		
	10024	addi \$sp,\$sp,8	#pop 2 items off stack		
	10028	jr \$ra	#return to caller		
	10032	L1: addi \$a0,\$a0,-1	#n >= 1: argument gets (n - 1)		
	10036	jal fact	#call fact with (n - 1)		
	10040	lw \$a0, 0(\$sp)	#return from jal: restore argument n		
	10044	lw \$ra, 4(\$sp)	#restore the return address		
<b>pc→</b>	10048	addi \$sp, \$sp, 8	#adjust stack pointer to pop 2 items		
	10052	mul \$v0,\$a0,\$v0	#return n * fact (n - 1)		
	10056	jr \$ra	#return to the caller		

	5056				
	5052	\$ra	8004	\$ra	10040
	5048	\$a0	4	\$a0	2
	5044	\$ra	10040	\$t0	1
<b>sp→</b>	5040	\$a0	3	\$v0	2
	5036	\$ra	10040		
	5032	\$a0	2		
	5028	\$ra	10040		
	5024	\$a0	1		
	5020	\$ra	10040		
	5016	\$a0	0		
	5012				
	5008				
	5004				
	5000				
	8000	jal fact			
	8004				
	...				
	10000	fact: addi \$sp, \$sp, -8	#adjust stack for 2 items		
	10004	sw \$ra, 4(\$sp)	#save the return address		
	10008	sw \$a0, 0(\$sp)	#save the argument n		
	10012	slti \$t0,\$a0,1	#test for n < 1		
	10016	beq \$t0,\$zero,L1	#if n >= 1, go to L1		
	10020	addi \$v0,\$zero,1	#return 1		
	10024	addi \$sp,\$sp,8	#pop 2 items off stack		
	10028	jr \$ra	#return to caller		
	10032	L1: addi \$a0,\$a0,-1	#n >= 1: argument gets (n - 1)		
	10036	jal fact	#call fact with (n - 1)		
	10040	lw \$a0, 0(\$sp)	#return from jal: restore argument n		
	10044	lw \$ra, 4(\$sp)	#restore the return address		
	10048	addi \$sp, \$sp, 8	#adjust stack pointer to pop 2 items		
<b>pc→</b>	10052	mul \$v0,\$a0,\$v0	#return n * fact (n - 1)		
	10056	jr \$ra	#return to the caller		

	5056				
	5052	\$ra	8004	\$ra	10040
	5048	\$a0	4	\$a0	2
	5044	\$ra	10040	\$t0	1
<b>sp→</b>	5040	\$a0	3	\$v0	2
	5036	\$ra	10040		
	5032	\$a0	2		
	5028	\$ra	10040		
	5024	\$a0	1		
	5020	\$ra	10040		
	5016	\$a0	0		
	5012				
	5008				
	5004				
	5000				
	8000	jal fact			
	8004				
	...				
	10000	fact: addi \$sp, \$sp, -8	#adjust stack for 2 items		
	10004	sw \$ra, 4(\$sp)	#save the return address		
	10008	sw \$a0, 0(\$sp)	#save the argument n		
	10012	slti \$t0,\$a0,1	#test for n < 1		
	10016	beq \$t0,\$zero,L1	#if n >= 1, go to L1		
	10020	addi \$v0,\$zero,1	#return 1		
	10024	addi \$sp,\$sp,8	#pop 2 items off stack		
	10028	jr \$ra	#return to caller		
	10032	L1: addi \$a0,\$a0,-1	#n >= 1: argument gets (n - 1)		
	10036	jal fact	#call fact with (n - 1)		
	10040	lw \$a0, 0(\$sp)	#return from jal: restore argument n		
	10044	lw \$ra, 4(\$sp)	#restore the return address		
	10048	addi \$sp, \$sp, 8	#adjust stack pointer to pop 2 items		
	10052	mul \$v0,\$a0,\$v0	#return n * fact (n - 1)		
<b>pc→</b>	10056	jr \$ra	#return to the caller		

	5056				
	5052	\$ra	8004	\$ra	10040
	5048	\$a0	4	\$a0	3
	5044	\$ra	10040	\$t0	1
<b>sp→</b>	5040	\$a0	3	\$v0	2
	5036	\$ra	10040		
	5032	\$a0	2		
	5028	\$ra	10040		
	5024	\$a0	1		
	5020	\$ra	10040		
	5016	\$a0	0		
	5012				
	5008				
	5004				
	5000				
	8000	jal fact			
	8004				
	...				
	10000	fact: addi \$sp, \$sp, -8	#adjust stack for 2 items		
	10004	sw \$ra, 4(\$sp)	#save the return address		
	10008	sw \$a0, 0(\$sp)	#save the argument n		
	10012	slti \$t0,\$a0,1	#test for n < 1		
	10016	beq \$t0,\$zero,L1	#if n >= 1, go to L1		
	10020	addi \$v0,\$zero,1	#return 1		
	10024	addi \$sp,\$sp,8	#pop 2 items off stack		
	10028	jr \$ra	#return to caller		
	10032	L1: addi \$a0,\$a0,-1	#n >= 1: argument gets (n - 1)		
	10036	jal fact	#call fact with (n - 1)		
<b>pc→</b>	10040	lw \$a0, 0(\$sp)	#return from jal: restore argument n		
	10044	lw \$ra, 4(\$sp)	#restore the return address		
	10048	addi \$sp, \$sp, 8	#adjust stack pointer to pop 2 items		
	10052	mul \$v0,\$a0,\$v0	#return n * fact (n - 1)		
	10056	jr \$ra	#return to the caller		



	5056				
	5052	\$ra	8004	\$ra	10040
	5048	\$a0	4	\$a0	3
	5044	\$ra	10040	\$t0	1
<b>sp→</b>	5040	\$a0	3	\$v0	2
	5036	\$ra	10040		
	5032	\$a0	2		
	5028	\$ra	10040		
	5024	\$a0	1		
	5020	\$ra	10040		
	5016	\$a0	0		
	5012				
	5008				
	5004				
	5000				
	8000	jal fact			
	8004				
	...				
	10000	fact: addi \$sp, \$sp, -8	#adjust stack for 2 items		
	10004	sw \$ra, 4(\$sp)	#save the return address		
	10008	sw \$a0, 0(\$sp)	#save the argument n		
	10012	slti \$t0,\$a0,1	#test for n < 1		
	10016	beq \$t0,\$zero,L1	#if n >= 1, go to L1		
	10020	addi \$v0,\$zero,1	#return 1		
	10024	addi \$sp,\$sp,8	#pop 2 items off stack		
	10028	jr \$ra	#return to caller		
	10032	L1: addi \$a0,\$a0,-1	#n >= 1: argument gets (n - 1)		
	10036	jal fact	#call fact with (n - 1)		
	10040	lw \$a0, 0(\$sp)	#return from jal: restore argument n		
<b>pc→</b>	10044	lw \$ra, 4(\$sp)	#restore the return address		
	10048	addi \$sp, \$sp, 8	#adjust stack pointer to pop 2 items		
	10052	mul \$v0,\$a0,\$v0	#return n * fact (n - 1)		
	10056	jr \$ra	#return to the caller		

	5056				
	5052	\$ra	8004	\$ra	10040
<b>sp→</b>	5048	\$a0	4	\$a0	3
	5044	\$ra	10040	\$t0	1
	5040	\$a0	3	\$v0	2
	5036	\$ra	10040		
	5032	\$a0	2		
	5028	\$ra	10040		
	5024	\$a0	1		
	5020	\$ra	10040		
	5016	\$a0	0		
	5012				
	5008				
	5004				
	5000				
	8000	jal fact			
	8004				
	...				
	10000	fact: addi \$sp, \$sp, -8	#adjust stack for 2 items		
	10004	sw \$ra, 4(\$sp)	#save the return address		
	10008	sw \$a0, 0(\$sp)	#save the argument n		
	10012	slti \$t0,\$a0,1	#test for n < 1		
	10016	beq \$t0,\$zero,L1	#if n >= 1, go to L1		
	10020	addi \$v0,\$zero,1	#return 1		
	10024	addi \$sp,\$sp,8	#pop 2 items off stack		
	10028	jr \$ra	#return to caller		
	10032	L1: addi \$a0,\$a0,-1	#n >= 1: argument gets (n - 1)		
	10036	jal fact	#call fact with (n - 1)		
	10040	lw \$a0, 0(\$sp)	#return from jal: restore argument n		
	10044	lw \$ra, 4(\$sp)	#restore the return address		
<b>pc→</b>	10048	addi \$sp, \$sp, 8	#adjust stack pointer to pop 2 items		
	10052	mul \$v0,\$a0,\$v0	#return n * fact (n - 1)		
	10056	jr \$ra	#return to the caller		

	5056				
	5052	\$ra	8004	\$ra	10040
sp→	5048	\$a0	4	\$a0	3
	5044	\$ra	10040	\$t0	1
	5040	\$a0	3	\$v0	6
	5036	\$ra	10040		
	5032	\$a0	2		
	5028	\$ra	10040		
	5024	\$a0	1		
	5020	\$ra	10040		
	5016	\$a0	0		
	5012				
	5008				
	5004				
	5000				
	8000	jal fact			
	8004				
	...				
	10000	fact: addi \$sp, \$sp, -8	#adjust stack for 2 items		
	10004	sw \$ra, 4(\$sp)	#save the return address		
	10008	sw \$a0, 0(\$sp)	#save the argument n		
	10012	slti \$t0,\$a0,1	#test for n < 1		
	10016	beq \$t0,\$zero,L1	#if n >= 1, go to L1		
	10020	addi \$v0,\$zero,1	#return 1		
	10024	addi \$sp,\$sp,8	#pop 2 items off stack		
	10028	jr \$ra	#return to caller		
	10032	L1: addi \$a0,\$a0,-1	#n >= 1: argument gets (n - 1)		
	10036	jal fact	#call fact with (n - 1)		
	10040	lw \$a0, 0(\$sp)	#return from jal: restore argument n		
	10044	lw \$ra, 4(\$sp)	#restore the return address		
	10048	addi \$sp, \$sp, 8	#adjust stack pointer to pop 2 items		
pc→	10052	mul \$v0,\$a0,\$v0	#return n * fact (n - 1)		
	10056	jr \$ra	#return to the caller		

	5056				
	5052	\$ra	8004	\$ra	10040
<b>sp→</b>	5048	\$a0	4	\$a0	3
	5044	\$ra	10040	\$t0	1
	5040	\$a0	3	\$v0	6
	5036	\$ra	10040		
	5032	\$a0	2		
	5028	\$ra	10040		
	5024	\$a0	1		
	5020	\$ra	10040		
	5016	\$a0	0		
	5012				
	5008				
	5004				
	5000				
	8000	jal fact			
	8004				
	...				
	10000	fact: addi \$sp, \$sp, -8	#adjust stack for 2 items		
	10004	sw \$ra, 4(\$sp)	#save the return address		
	10008	sw \$a0, 0(\$sp)	#save the argument n		
	10012	slti \$t0,\$a0,1	#test for n < 1		
	10016	beq \$t0,\$zero,L1	#if n >= 1, go to L1		
	10020	addi \$v0,\$zero,1	#return 1		
	10024	addi \$sp,\$sp,8	#pop 2 items off stack		
	10028	jr \$ra	#return to caller		
	10032	L1: addi \$a0,\$a0,-1	#n >= 1: argument gets (n - 1)		
	10036	jal fact	#call fact with (n - 1)		
	10040	lw \$a0, 0(\$sp)	#return from jal: restore argument n		
	10044	lw \$ra, 4(\$sp)	#restore the return address		
	10048	addi \$sp, \$sp, 8	#adjust stack pointer to pop 2 items		
	10052	mul \$v0,\$a0,\$v0	#return n * fact (n - 1)		
<b>pc→</b>	10056	jr \$ra	#return to the caller		

	5056				
	5052	\$ra	8004	\$ra	10040
<b>sp→</b>	5048	\$a0	4	\$a0	4
	5044	\$ra	10040	\$t0	1
	5040	\$a0	3	\$v0	6
	5036	\$ra	10040		
	5032	\$a0	2		
	5028	\$ra	10040		
	5024	\$a0	1		
	5020	\$ra	10040		
	5016	\$a0	0		
	5012				
	5008				
	5004				
	5000				
	8000	jal fact			
	8004				
	...				
	10000	fact: addi \$sp, \$sp, -8	#adjust stack for 2 items		
	10004	sw \$ra, 4(\$sp)	#save the return address		
	10008	sw \$a0, 0(\$sp)	#save the argument n		
	10012	slti \$t0,\$a0,1	#test for n < 1		
	10016	beq \$t0,\$zero,L1	#if n >= 1, go to L1		
	10020	addi \$v0,\$zero,1	#return 1		
	10024	addi \$sp,\$sp,8	#pop 2 items off stack		
	10028	jr \$ra	#return to caller		
	10032	L1: addi \$a0,\$a0,-1	#n >= 1: argument gets (n - 1)		
	10036	jal fact	#call fact with (n - 1)		
<b>pc→</b>	10040	lw \$a0, 0(\$sp)	#return from jal: restore argument n		
	10044	lw \$ra, 4(\$sp)	#restore the return address		
	10048	addi \$sp, \$sp, 8	#adjust stack pointer to pop 2 items		
	10052	mul \$v0,\$a0,\$v0	#return n * fact (n - 1)		
	10056	jr \$ra	#return to the caller		

	5056				
	5052	\$ra	8004	\$ra	10040
<b>sp→</b>	5048	\$a0	4	\$a0	4
	5044	\$ra	10040	\$t0	1
	5040	\$a0	3	\$v0	6
	5036	\$ra	10040		
	5032	\$a0	2		
	5028	\$ra	10040		
	5024	\$a0	1		
	5020	\$ra	10040		
	5016	\$a0	0		
	5012				
	5008				
	5004				
	5000				
	8000	jal fact			
	8004				
	...				
	10000	fact: addi \$sp, \$sp, -8	#adjust stack for 2 items		
	10004	sw \$ra, 4(\$sp)	#save the return address		
	10008	sw \$a0, 0(\$sp)	#save the argument n		
	10012	slti \$t0,\$a0,1	#test for n < 1		
	10016	beq \$t0,\$zero,L1	#if n >= 1, go to L1		
	10020	addi \$v0,\$zero,1	#return 1		
	10024	addi \$sp,\$sp,8	#pop 2 items off stack		
	10028	jr \$ra	#return to caller		
	10032	L1: addi \$a0,\$a0,-1	#n >= 1: argument gets (n - 1)		
	10036	jal fact	#call fact with (n - 1)		
	10040	lw \$a0, 0(\$sp)	#return from jal: restore argument n		
<b>pc→</b>	10044	lw \$ra, 4(\$sp)	#restore the return address		
	10048	addi \$sp, \$sp, 8	#adjust stack pointer to pop 2 items		
	10052	mul \$v0,\$a0,\$v0	#return n * fact (n - 1)		
	10056	jr \$ra	#return to the caller		

<b>sp→</b>	5056				
	5052	\$ra	8004	\$ra	8004
	5048	\$a0	4	\$a0	4
	5044	\$ra	10040	\$t0	1
	5040	\$a0	3	\$v0	6
	5036	\$ra	10040		
	5032	\$a0	2		
	5028	\$ra	10040		
	5024	\$a0	1		
	5020	\$ra	10040		
	5016	\$a0	0		
	5012				
	5008				
	5004				
	5000				
	8000	jal fact			
	8004				
	...				
	10000	fact: addi \$sp, \$sp, -8	#adjust stack for 2 items		
	10004	sw \$ra, 4(\$sp)	#save the return address		
	10008	sw \$a0, 0(\$sp)	#save the argument n		
	10012	slti \$t0,\$a0,1	#test for n < 1		
	10016	beq \$t0,\$zero,L1	#if n >= 1, go to L1		
	10020	addi \$v0,\$zero,1	#return 1		
	10024	addi \$sp,\$sp,8	#pop 2 items off stack		
	10028	jr \$ra	#return to caller		
	10032	L1: addi \$a0,\$a0,-1	#n >= 1: argument gets (n - 1)		
	10036	jal fact	#call fact with (n - 1)		
	10040	lw \$a0, 0(\$sp)	#return from jal: restore argument n		
	10044	lw \$ra, 4(\$sp)	#restore the return address		
<b>pc→</b>	10048	addi \$sp, \$sp, 8	#adjust stack pointer to pop 2 items		
	10052	mul \$v0,\$a0,\$v0	#return n * fact (n - 1)		
	10056	jr \$ra	#return to the caller		

<b>sp→</b>	5056				
	5052	\$ra	8004	\$ra	8004
	5048	\$a0	4	\$a0	4
	5044	\$ra	10040	\$t0	1
	5040	\$a0	3	\$v0	24
	5036	\$ra	10040		
	5032	\$a0	2		
	5028	\$ra	10040		
	5024	\$a0	1		
	5020	\$ra	10040		
	5016	\$a0	0		
	5012				
	5008				
	5004				
	5000				
	8000	jal fact			
	8004				
	...				
	10000	fact: addi \$sp, \$sp, -8	#adjust stack for 2 items		
	10004	sw \$ra, 4(\$sp)	#save the return address		
	10008	sw \$a0, 0(\$sp)	#save the argument n		
	10012	slti \$t0,\$a0,1	#test for n < 1		
	10016	beq \$t0,\$zero,L1	#if n >= 1, go to L1		
	10020	addi \$v0,\$zero,1	#return 1		
	10024	addi \$sp,\$sp,8	#pop 2 items off stack		
	10028	jr \$ra	#return to caller		
	10032	L1: addi \$a0,\$a0,-1	#n >= 1: argument gets (n - 1)		
	10036	jal fact	#call fact with (n - 1)		
	10040	lw \$a0, 0(\$sp)	#return from jal: restore argument n		
	10044	lw \$ra, 4(\$sp)	#restore the return address		
	10048	addi \$sp, \$sp, 8	#adjust stack pointer to pop 2 items		
<b>pc→</b>	10052	mul \$v0,\$a0,\$v0	#return n * fact (n - 1)		
	10056	jr \$ra	#return to the caller		



sp→	5056				
	5052	\$ra	8004	\$ra	8004
	5048	\$a0	4	\$a0	4
	5044	\$ra	10040	\$t0	1
	5040	\$a0	3	\$v0	24
	5036	\$ra	10040		
	5032	\$a0	2		
	5028	\$ra	10040		
	5024	\$a0	1		
	5020	\$ra	10040		
	5016	\$a0	0		
	5012				
	5008				
	5004				
	5000				
	8000	jal fact			
	8004				
	...				
	10000	fact: addi \$sp, \$sp, -8	#adjust stack for 2 items		
	10004	sw \$ra, 4(\$sp)	#save the return address		
	10008	sw \$a0, 0(\$sp)	#save the argument n		
	10012	slti \$t0,\$a0,1	#test for n < 1		
	10016	beq \$t0,\$zero,L1	#if n >= 1, go to L1		
	10020	addi \$v0,\$zero,1	#return 1		
	10024	addi \$sp,\$sp,8	#pop 2 items off stack		
	10028	jr \$ra	#return to caller		
	10032	L1: addi \$a0,\$a0,-1	#n >= 1: argument gets (n - 1)		
	10036	jal fact	#call fact with (n - 1)		
	10040	lw \$a0, 0(\$sp)	#return from jal: restore argument n		
	10044	lw \$ra, 4(\$sp)	#restore the return address		
	10048	addi \$sp, \$sp, 8	#adjust stack pointer to pop 2 items		
	10052	mul \$v0,\$a0,\$v0	#return n * fact (n - 1)		
pc→	10056	jr \$ra	#return to the caller		

<b>sp→</b>	5056				
	5052	\$ra	8004	\$ra	8004
	5048	\$a0	4	\$a0	4
	5044	\$ra	10040	\$t0	1
	5040	\$a0	3	\$v0	24
	5036	\$ra	10040		
	5032	\$a0	2		
	5028	\$ra	10040		
	5024	\$a0	1		
	5020	\$ra	10040		
	5016	\$a0	0		
	5012				
	5008				
	5004				
	5000				
	8000	jal fact			
<b>pc→</b>	8004				
	...				
	10000	fact: addi \$sp, \$sp, -8	#adjust stack for 2 items		
	10004	sw \$ra, 4(\$sp)	#save the return address		
	10008	sw \$a0, 0(\$sp)	#save the argument n		
	10012	slti \$t0,\$a0,1	#test for n < 1		
	10016	beq \$t0,\$zero,L1	#if n >= 1, go to L1		
	10020	addi \$v0,\$zero,1	#return 1		
	10024	addi \$sp,\$sp,8	#pop 2 items off stack		
	10028	jr \$ra	#return to caller		
	10032	L1: addi \$a0,\$a0,-1	#n >= 1: argument gets (n - 1)		
	10036	jal fact	#call fact with (n - 1)		
	10040	lw \$a0, 0(\$sp)	#return from jal: restore argument n		
	10044	lw \$ra, 4(\$sp)	#restore the return address		
	10048	addi \$sp, \$sp, 8	#adjust stack pointer to pop 2 items		
	10052	mul \$v0,\$a0,\$v0	#return n * fact (n - 1)		
	10056	jr \$ra	#return to the caller		