(1) If necessary, f pushes EAX, ECX, EDX, and parameters:

```assembly
pushl %ecx
pushl param3
pushl param2
pushl param1
```

(2) f executes call instruction:

```assembly
call g
```

(3) g pushes f’s EBP:

```assembly
pushl %ebp
```

(4) g copies ESP to EBP:

```assembly
movl %esp,%ebp
```

(5) If necessary, g pushes EBX, ESI, EDI and local variables:

```assembly
pushl %ebx
pushl localvar1
pushl localvar2
pushl localvar3
```

When function f calls function g:
(6) g uses parameters and local variables to compute a return value, and moves that value to EAX:

movl 8(%ebp), ???
movl -4(%ebp), %ebx
movl -8(%ebp), ???
movl -12(%ebp), ???
movl ???, %eax

(7) If necessary, g restores EBX, ESI, EDI

movl 8(%ebp), ???
movl 12(%ebp), ???

(8) g copies EBP to ESP:

movl %ebp,%esp

(9) g pops from stack into EBP:

popl %ebp

(10) g executes ret instruction:

ret

(11) f pops actual parameters from stack, and uses return value in EAX:

addl $12, %esp
movl %eax, ???

(12) If necessary, f restores EAX, ECX, EDX

popl %ecx

Epilog