```
 1    StubRoutines::call_stub [0xa75af477, 0xa75af548[ (209 bytes)
 2      0xa75af477: push    %ebp
 3      0xa75af478: mov     %esp,%ebp
 4      0xa75af47a: mov     0x20(%ebp),%ecx
 5      0xa75af47d: shl     $0x2,%ecx
 6      0xa75af480: add     $0x10,%ecx
 7      0xa75af483: sub     %ecx,%esp
 8      0xa75af485: and     $0xfffffff0,%esp
 9      0xa75af488: mov     %edi,-0x4(%ebp)
10      0xa75af48b: mov     %esi,-0x8(%ebp)
11      0xa75af48e: mov     %ebx,-0xc(%ebp)
12      0xa75af491: stmxcsr -0x10(%ebp)
13      0xa75af495: mov     -0x10(%ebp),%eax
14      0xa75af498: and     $0xffc0,%eax
15      0xa75af49e: cmp     0xb792d12c,%eax
16      0xa75af4a4: je      0xa75af4b1
17      0xa75af4aa: ldmxcsr 0xb792d12c
18      0xa75af4b1: fldcw   0xb792d11c
19      0xa75af4b7: mov     0x24(%ebp),%ecx
20      0xa75af4ba: cmpl    $0x0,0x4(%ecx)
21      0xa75af4c1: je      0xa75af4d8
22      0xa75af4c7: push    $0xb75ae8b8
23      0xa75af4cc: call    0xa75af4d1
24      0xa75af4d1: pusha
25      0xa75af4d2: call    0xb7052f32
26      0xa75af4d7: hlt
27      ;; pass parameters if any
28      0xa75af4d8: mov     0x20(%ebp),%ecx
29      0xa75af4db: test    %ecx,%ecx
30      0xa75af4dd: je      0xa75af4f3
31      0xa75af4e3: mov     0x1c(%ebp),%edx
32      0xa75af4e6: xor     %ebx,%ebx
33      ;; loop:
34      0xa75af4e8: mov     -0x4(%edx,%ecx,4),%eax
35      0xa75af4ec: mov     %eax,(%esp,%ebx,4)
36      0xa75af4ef: inc     %ebx
37      0xa75af4f0: dec     %ecx
38      0xa75af4f1: jne     0xa75af4e8
39      ;; parameters_done:
40      0xa75af4f3: mov     0x14(%ebp),%ebx
41      0xa75af4f6: mov     0x18(%ebp),%eax
42      0xa75af4f9: mov     %esp,%esi
43      ;; call Java function
44      0xa75af4fb: call    *%eax
45      ;; call_stub_return_address:
46      ;; L_skip:
47      0xa75af4fd: mov     0xc(%ebp),%edi
48      0xa75af500: mov     0x10(%ebp),%esi
49      0xa75af503: cmp     $0xb,%esi
50      0xa75af506: je      0xa75af535
51      0xa75af50c: cmp     $0x6,%esi
52      0xa75af50f: je      0xa75af53c
53      0xa75af515: cmp     $0x7,%esi
54      0xa75af518: je      0xa75af542
55      0xa75af51e: mov     %eax,(%edi)
56      ;; exit:
57      0xa75af520: lea     -0x10(%ebp),%esp
58      0xa75af523: ldmxcsr -0x10(%ebp)
59      0xa75af527: mov     -0xc(%ebp),%ebx
60      0xa75af52a: mov     -0x8(%ebp),%esi
61      0xa75af52d: mov     -0x4(%ebp),%edi
62      0xa75af530: add     $0x10,%esp
63      0xa75af533: pop     %ebp
64      0xa75af534: ret
65      ;; is_long:
66      0xa75af535: mov     %eax,(%edi)
67      0xa75af537: mov     %edx,0x4(%edi)
68      0xa75af53a: jmp     0xa75af520
69      ;; is_float:
70      0xa75af53c: vmovss %xmm0,(%edi)
71      0xa75af540: jmp     0xa75af520
72      ;; is_double:
73      0xa75af542: vmovsd %xmm0,(%edi)
74      0xa75af546: jmp     0xa75af520
75
```