

```
1  zbhuang@zbhuang-VirtualBox:~/OpenJDK/cppapps$ gcc -g functionf.c -o functionf.o
2  zbhuang@zbhuang-VirtualBox:~/OpenJDK/cppapps$ gdb -q functionf.o | tee functionf.asm.s
3  Reading symbols from functionf.o...done.
4  (gdb) disassemble /m main
5  Dump of assembler code for function main:
6  11      {
7      0x00000000004004f5 <+0>:      push    %rbp
8      0x00000000004004f6 <+1>:      mov     %rsp,%rbp
9      0x00000000004004f9 <+4>:      sub     $0x20,%rsp
10     0x00000000004004fd <+8>:      mov     %edi,-0x14(%rbp)
11     0x0000000000400500 <+11>:     mov     %rsi,-0x20(%rbp)
12
13     12      int i = 77;
14     0x0000000000400504 <+15>:     movl    $0x4d,-0x4(%rbp)
15
16     13      i = f(i, 8);
17     0x000000000040050b <+22>:     mov     -0x4(%rbp),%eax
18     0x000000000040050e <+25>:     mov     $0x8,%esi
19     0x0000000000400513 <+30>:     mov     %eax,%edi
20     0x0000000000400515 <+32>:     callq   0x4004d6 <f>
21     0x000000000040051a <+37>:     mov     %eax,-0x4(%rbp)
22
23     14      i %= 5;
24     0x000000000040051d <+40>:     mov     -0x4(%rbp),%ecx
25     0x0000000000400520 <+43>:     mov     $0x66666667,%edx
26     0x0000000000400525 <+48>:     mov     %ecx,%eax
27     0x0000000000400527 <+50>:     imul    %edx
28     0x0000000000400529 <+52>:     sar     %edx
29     0x000000000040052b <+54>:     mov     %ecx,%eax
30     0x000000000040052d <+56>:     sar     $0x1f,%eax
31     0x0000000000400530 <+59>:     sub     %eax,%edx
32     0x0000000000400532 <+61>:     mov     %edx,%eax
33     0x0000000000400534 <+63>:     shl     $0x2,%eax
34     0x0000000000400537 <+66>:     add     %edx,%eax
35     0x0000000000400539 <+68>:     sub     %eax,%ecx
36     0x000000000040053b <+70>:     mov     %ecx,%eax
37     0x000000000040053d <+72>:     mov     %eax,-0x4(%rbp)
38
39     15      return i;
40     0x0000000000400540 <+75>:     mov     -0x4(%rbp),%eax
41
42     16      }
43     0x0000000000400543 <+78>:     leaveq  %eax
44     0x0000000000400544 <+79>:     retq
45
46  End of assembler dump.
47  (gdb) disassemble /m f
48  Dump of assembler code for function f:
49  3      {
50     0x00000000004004d6 <+0>:      push    %rbp
51     0x00000000004004d7 <+1>:      mov     %rsp,%rbp
52     0x00000000004004da <+4>:      mov     %edi,-0x14(%rbp)
53     0x00000000004004dd <+7>:      mov     %esi,-0x18(%rbp)
54
55     4      int i, j;
56     5      i = x;
57     0x00000000004004e0 <+10>:     mov     -0x14(%rbp),%eax
58     0x00000000004004e3 <+13>:     mov     %eax,-0x8(%rbp)
59
60     6      j = i * y;
61     0x00000000004004e6 <+16>:     mov     -0x8(%rbp),%eax
62     0x00000000004004e9 <+19>:     imul    -0x18(%rbp),%eax
63     0x00000000004004ed <+23>:     mov     %eax,-0x4(%rbp)
64
65     7      return j;
66     0x00000000004004f0 <+26>:     mov     -0x4(%rbp),%eax
67
68     8      }
69     0x00000000004004f3 <+29>:     pop     %rbp
70     0x00000000004004f4 <+30>:     retq
71
72  End of assembler dump.
73  (gdb) run
74  Starting program: /home/zbhuang/OpenJDK/cppapps/functionf.o
75  [Inferior 1 (process 2844) exited with code 01]
76  (gdb) quit
77
```