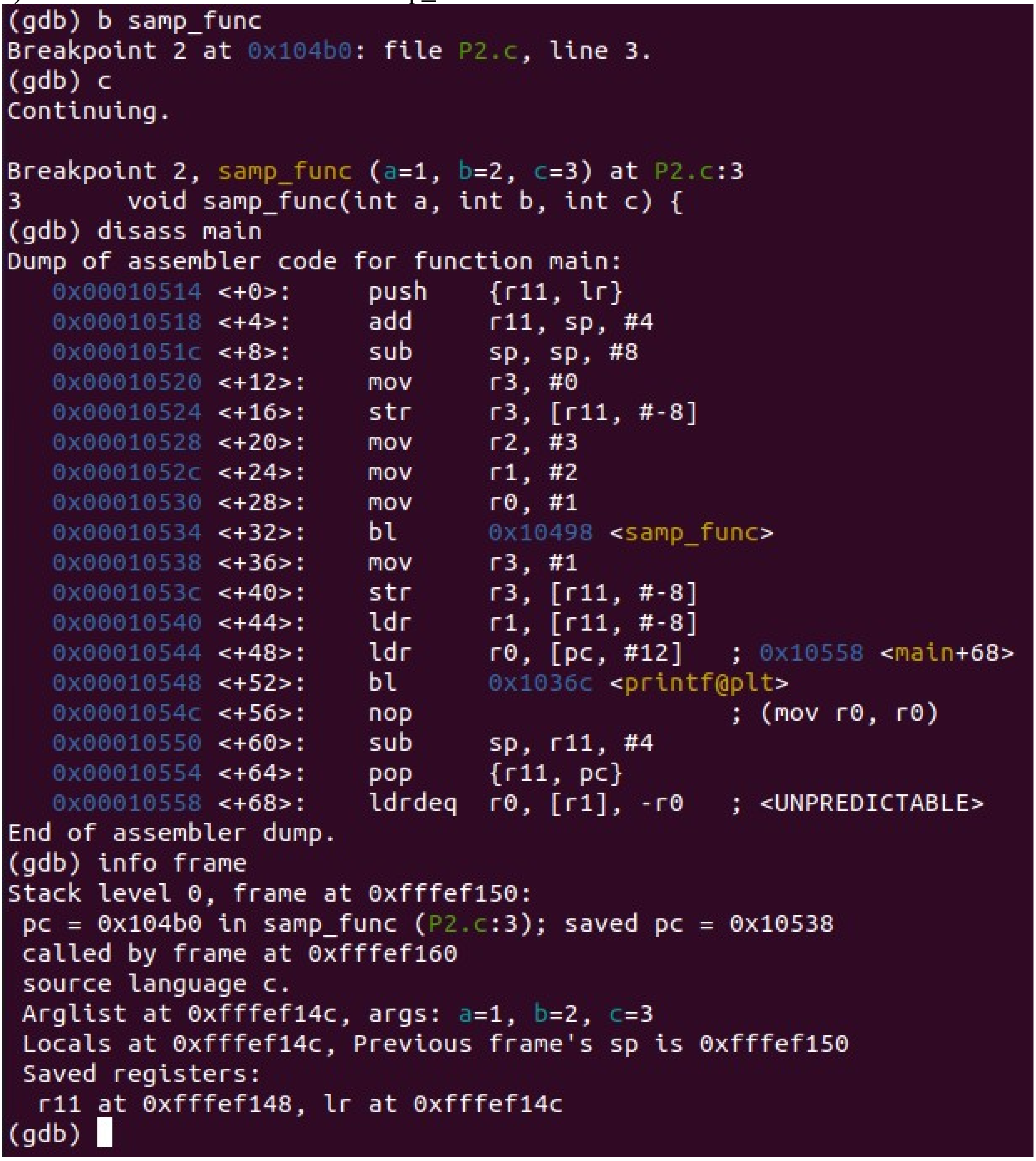
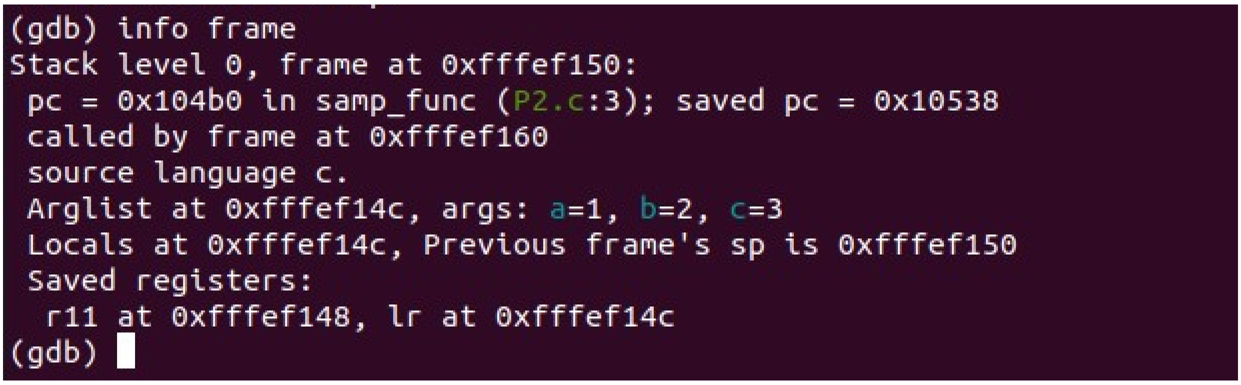
Q1) Return Address of the function samp_func is **0x00010538**.



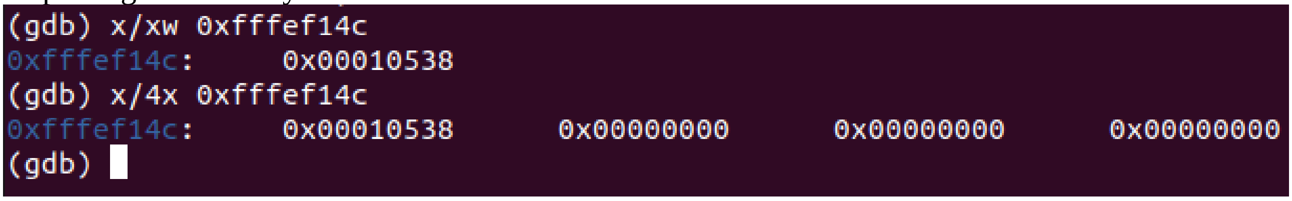
```
(gdb) b samp_func
Breakpoint 2 at 0x104b0: file P2.c, line 3.
(gdb) c
Continuing.

Breakpoint 2, samp_func (a=1, b=2, c=3) at P2.c:3
3       void samp_func(int a, int b, int c) {
(gdb) disass main
Dump of assembler code for function main:
   0x00010514 <+0>:     push    {r11, lr}
   0x00010518 <+4>:     add     r11, sp, #4
   0x0001051c <+8>:     sub     sp, sp, #8
   0x00010520 <+12>:    mov     r3, #0
   0x00010524 <+16>:    str     r3, [r11, #-8]
   0x00010528 <+20>:    mov     r2, #3
   0x0001052c <+24>:    mov     r1, #2
   0x00010530 <+28>:    mov     r0, #1
   0x00010534 <+32>:    bl      0x10498 <samp_func>
   0x00010538 <+36>:    mov     r3, #1
   0x0001053c <+40>:    str     r3, [r11, #-8]
   0x00010540 <+44>:    ldr     r1, [r11, #-8]
   0x00010544 <+48>:    ldr     r0, [pc, #12]   ; 0x10558 <main+68>
   0x00010548 <+52>:    bl      0x1036c <printf@plt>
   0x0001054c <+56>:    nop                     ; (mov r0, r0)
   0x00010550 <+60>:    sub     sp, r11, #4
   0x00010554 <+64>:    pop     {r11, pc}
   0x00010558 <+68>:    ldrdeq  r0, [r1], -r0   ; <UNPREDICTABLE>
End of assembler dump.
(gdb) info frame
Stack level 0, frame at 0xfffef150:
 pc = 0x104b0 in samp_func (P2.c:3); saved pc = 0x10538
 called by frame at 0xfffef160
 source language c.
 Arglist at 0xfffef14c, args: a=1, b=2, c=3
 Locals at 0xfffef14c, Previous frame's sp is 0xfffef150
 Saved registers:
  r11 at 0xfffef148, lr at 0xfffef14c
(gdb) 
```

Q2) Return address of samp_func is stored at the address **0xfffef14c** on the stack (lr address).



```
(gdb) info frame
Stack level 0, frame at 0xfffef150:
 pc = 0x104b0 in samp_func (P2.c:3); saved pc = 0x10538
 called by frame at 0xfffef160
 source language c.
 Arglist at 0xfffef14c, args: a=1, b=2, c=3
 Locals at 0xfffef14c, Previous frame's sp is 0xfffef150
 Saved registers:
  r11 at 0xfffef148, lr at 0xfffef14c
(gdb) 
```

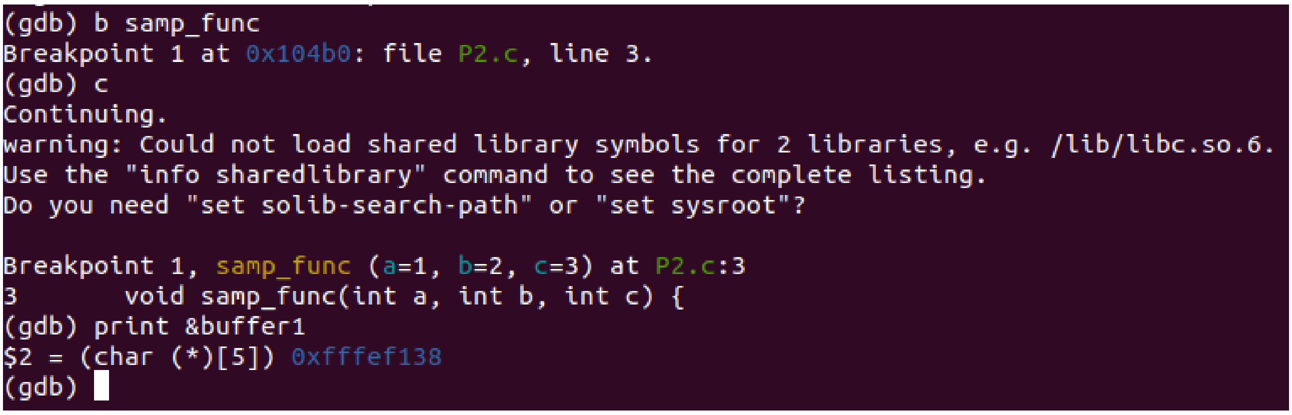
Inspecting the memory at the address 0xfffef14c for verification:



```
(gdb) x/xw 0xfffef14c
0xfffef14c:     0x00010538
(gdb) x/4x 0xfffef14c
0xfffef14c:     0x00010538      0x00000000      0x00000000      0x00000000
(gdb) 
```

It has 0x00010538, which is the return address of samp_func.

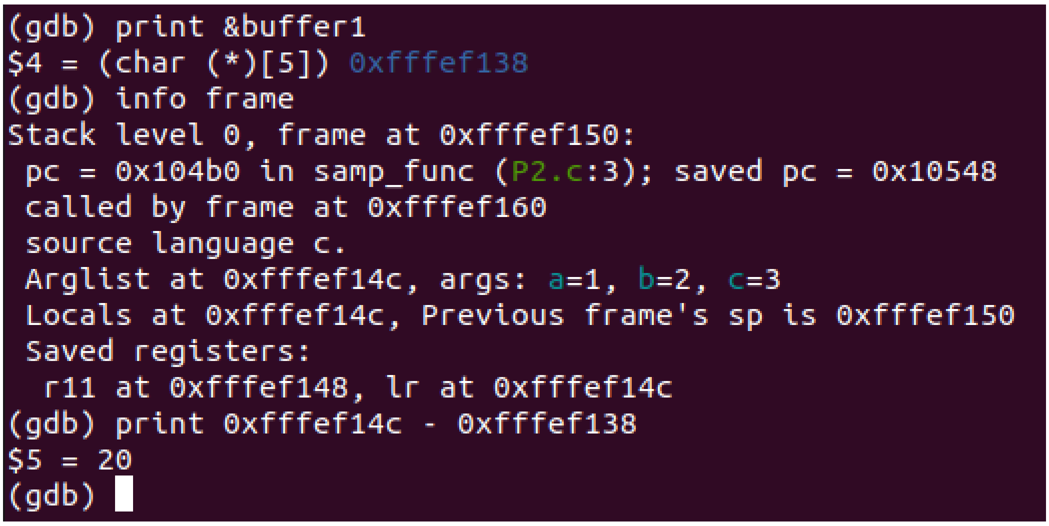
Q3) buffer1 is stored on the stack at the address **0xfffef138**



```
(gdb) b samp_func
Breakpoint 1 at 0x104b0: file P2.c, line 3.
(gdb) c
Continuing.
warning: Could not load shared library symbols for 2 libraries, e.g. /lib/libc.so.6.
Use the "info sharedlibrary" command to see the complete listing.
Do you need "set solib-search-path" or "set sysroot"?

Breakpoint 1, samp_func (a=1, b=2, c=3) at P2.c:3
3       void samp_func(int a, int b, int c) {
(gdb) print &buffer1
$2 = (char (*)[5]) 0xfffef138
(gdb) 
```

Q4) **20 bytes**

Address of buffer1: *print &buffer1* : **0xfffef130**
Return Address of samp_func: &lr : **0xfffef14c**
Distance = Address of Return Address – Address of Buffer1

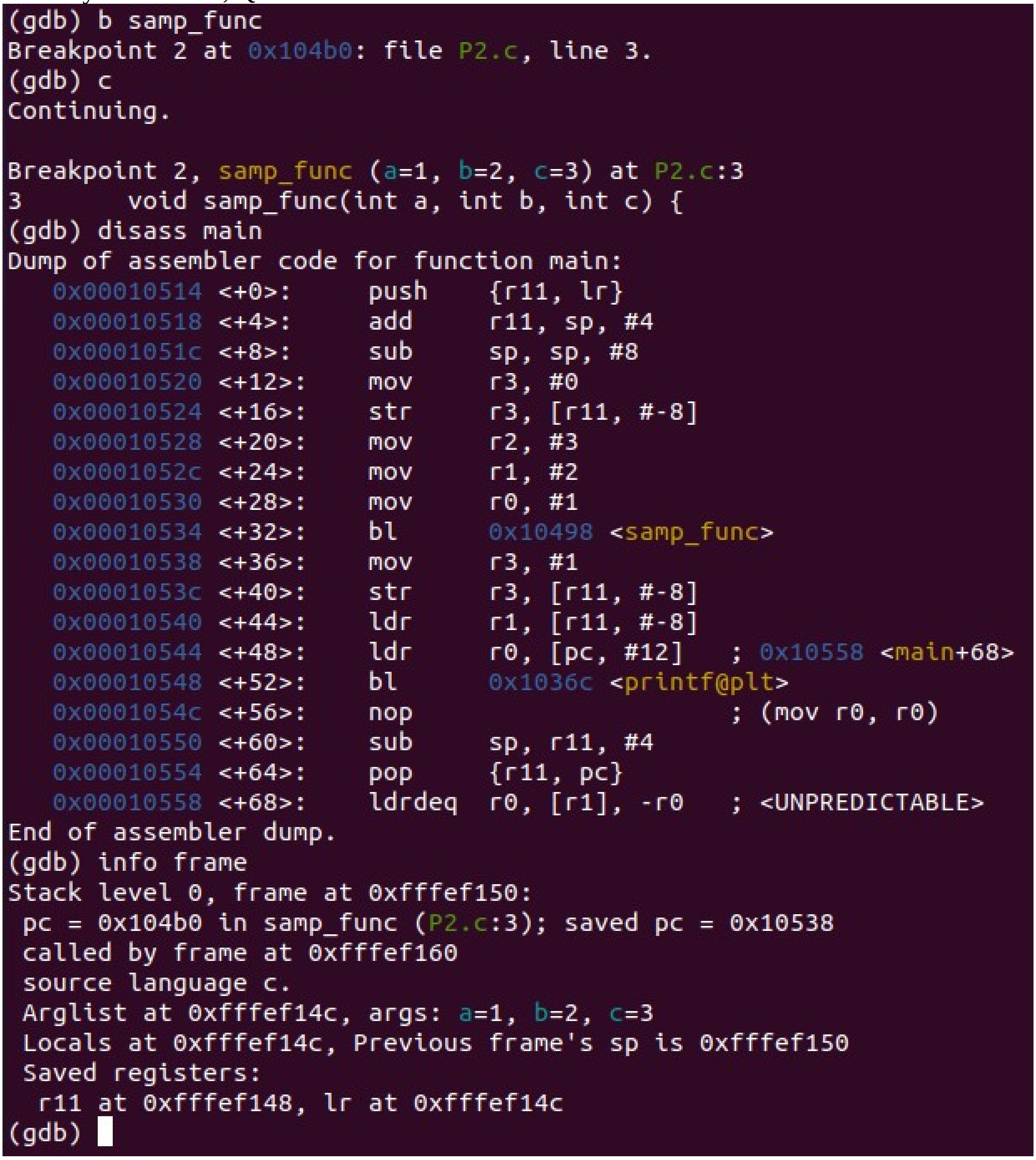


```
(gdb) print &buffer1
$4 = (char (*)[5]) 0xfffef138
(gdb) info frame
Stack level 0, frame at 0xfffef150:
 pc = 0x104b0 in samp_func (P2.c:3); saved pc = 0x10548
 called by frame at 0xfffef160
 source language c.
 Arglist at 0xfffef14c, args: a=1, b=2, c=3
 Locals at 0xfffef14c, Previous frame's sp is 0xfffef150
 Saved registers:
  r11 at 0xfffef148, lr at 0xfffef14c
(gdb) print 0xfffef14c - 0xfffef138
$5 = 20
(gdb) 
```

0xfffef14c – 0xfffef138=20 bytes

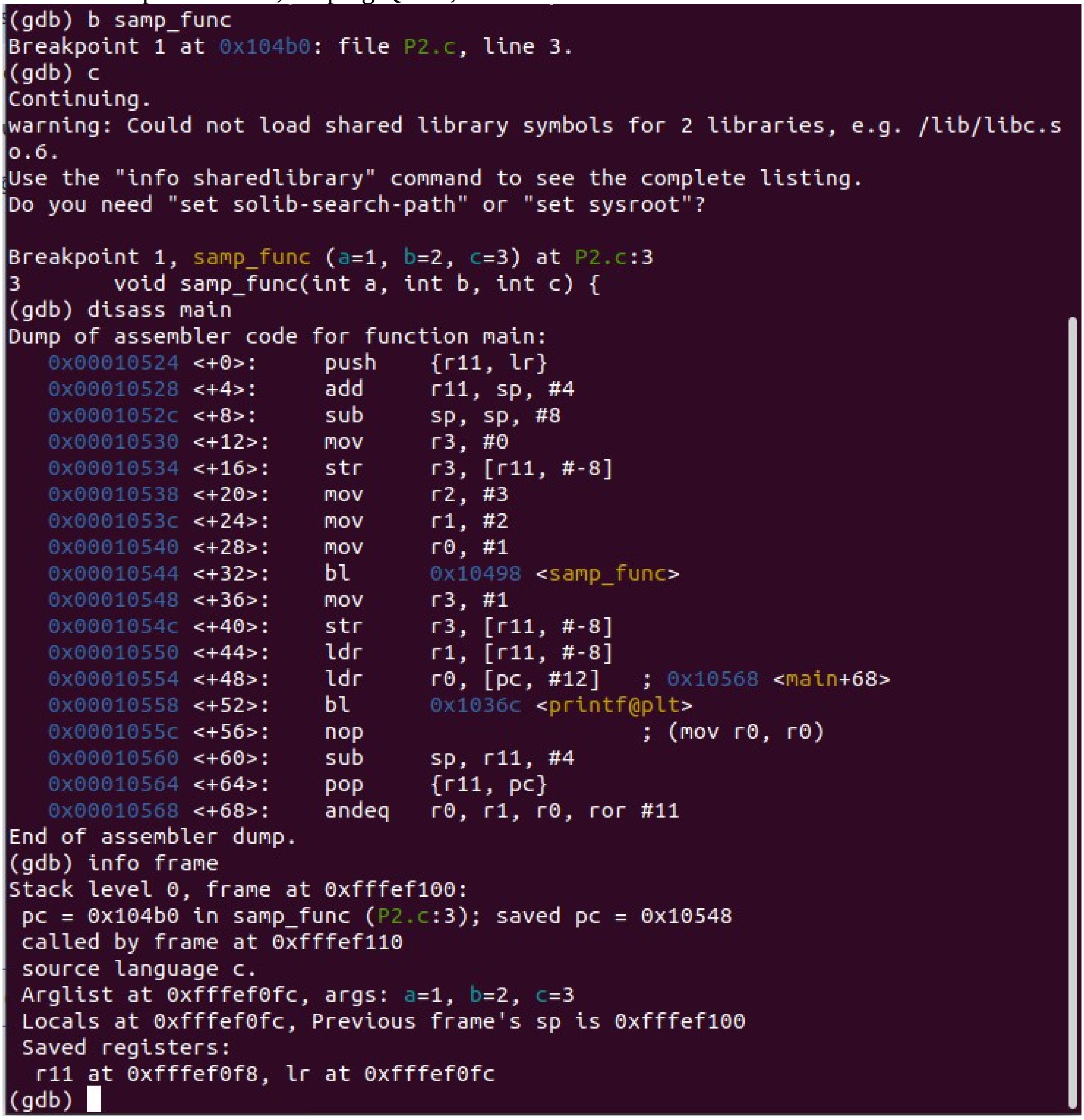
Q5) **P = 20**, the difference calculated in the previous question (offset).
Q = 16

Previously when P = 0, Q = 0:



```
(gdb) b samp_func
Breakpoint 2 at 0x104b0: file P2.c, line 3.
(gdb) c
Continuing.

Breakpoint 2, samp_func (a=1, b=2, c=3) at P2.c:3
3       void samp_func(int a, int b, int c) {
(gdb) disass main
Dump of assembler code for function main:
   0x00010514 <+0>:     push    {r11, lr}
   0x00010518 <+4>:     add     r11, sp, #4
   0x0001051c <+8>:     sub     sp, sp, #8
   0x00010520 <+12>:    mov     r3, #0
   0x00010524 <+16>:    str     r3, [r11, #-8]
   0x00010528 <+20>:    mov     r2, #3
   0x0001052c <+24>:    mov     r1, #2
   0x00010530 <+28>:    mov     r0, #1
   0x00010534 <+32>:    bl      0x10498 <samp_func>
   0x00010538 <+36>:    mov     r3, #1
   0x0001053c <+40>:    str     r3, [r11, #-8]
   0x00010540 <+44>:    ldr     r1, [r11, #-8]
   0x00010544 <+48>:    ldr     r0, [pc, #12]   ; 0x10558 <main+68>
   0x00010548 <+52>:    bl      0x1036c <printf@plt>
   0x0001054c <+56>:    nop                     ; (mov r0, r0)
   0x00010550 <+60>:    sub     sp, r11, #4
   0x00010554 <+64>:    pop     {r11, pc}
   0x00010558 <+68>:    ldrdeq  r0, [r1], -r0   ; <UNPREDICTABLE>
End of assembler dump.
(gdb) info frame
Stack level 0, frame at 0xfffef150:
 pc = 0x104b0 in samp_func (P2.c:3); saved pc = 0x10538
 called by frame at 0xfffef160
 source language c.
 Arglist at 0xfffef14c, args: a=1, b=2, c=3
 Locals at 0xfffef14c, Previous frame's sp is 0xfffef150
 Saved registers:
  r11 at 0xfffef148, lr at 0xfffef14c
(gdb) 
```

Now as we update P = 20, keeping Q as 0,



```
(gdb) b samp_func
Breakpoint 1 at 0x104b0: file P2.c, line 3.
(gdb) c
Continuing.
warning: Could not load shared library symbols for 2 libraries, e.g. /lib/libc.s
o.6.
Use the "info sharedlibrary" command to see the complete listing.
Do you need "set solib-search-path" or "set sysroot"?

Breakpoint 1, samp_func (a=1, b=2, c=3) at P2.c:3
3       void samp_func(int a, int b, int c) {
(gdb) disass main
Dump of assembler code for function main:
   0x00010524 <+0>:     push    {r11, lr}
   0x00010528 <+4>:     add     r11, sp, #4
   0x0001052c <+8>:     sub     sp, sp, #8
   0x00010530 <+12>:    mov     r3, #0
   0x00010534 <+16>:    str     r3, [r11, #-8]
   0x00010538 <+20>:    mov     r2, #3
   0x0001053c <+24>:    mov     r1, #2
   0x00010540 <+28>:    mov     r0, #1
   0x00010544 <+32>:    bl      0x10498 <samp_func>
   0x00010548 <+36>:    mov     r3, #1
   0x0001054c <+40>:    str     r3, [r11, #-8]
   0x00010550 <+44>:    ldr     r1, [r11, #-8]
   0x00010554 <+48>:    ldr     r0, [pc, #12]   ; 0x10568 <main+68>
   0x00010558 <+52>:    bl      0x1036c <printf@plt>
   0x0001055c <+56>:    nop                     ; (mov r0, r0)
   0x00010560 <+60>:    sub     sp, r11, #4
   0x00010564 <+64>:    pop     {r11, pc}
   0x00010568 <+68>:    andeq   r0, r1, r0, ror #11
End of assembler dump.
(gdb) info frame
Stack level 0, frame at 0xfffef100:
 pc = 0x104b0 in samp_func (P2.c:3); saved pc = 0x10548
 called by frame at 0xfffef110
 source language c.
 Arglist at 0xfffef0fc, args: a=1, b=2, c=3
 Locals at 0xfffef0fc, Previous frame's sp is 0xfffef100
 Saved registers:
  r11 at 0xfffef0f8, lr at 0xfffef0fc
(gdb) 
```

Return Address of the function samp_func now is **0x00010548**.

Hence, **0x00010548 - 0x00010538 = 0x10**
Which is 16 in decimal.

P = 20
Q = 16
This will make line 12 to be skipped from execution.