

Q1)

```
0x00010510 <+120>: mov     r3, r0
0x00010514 <+124>: str     r3, [r11, #-36] ; 0xffffffffdc
=> 0x00010518 <+128>: ldr     r3, [r11, #-32] ; 0xfffffffffe0
0x0001051c <+132>: add     r3, r3, #1
0x00010520 <+136>: str     r3, [r11, #-32] ; 0xfffffffffe0
0x00010524 <+140>: ldr     r3, [r11, #-32] ; 0xfffffffffe0
0x00010528 <+144>: cmp     r3, #4
0x0001052c <+148>: ble     0x104dc <main+68>
0x00010530 <+152>: ldr     r0, [r11, #-36] ; 0xffffffffdc
0x00010534 <+156>: bl      0x10884 <__extendsfdf2>
0x00010538 <+160>: mov     r2, r0
0x0001053c <+164>: mov     r3, r1
0x00010540 <+168>: ldr     r0, [pc, #52] ; 0x1057c <main+228>
--Type <RET> for more, q to quit, c to continue without paging--q
Quit
(gdb) b *0x0001051c
Breakpoint 3 at 0x1051c: file P3.c, line 7.
(gdb) jump *0x0001051c
Continuing at 0x1051c.

Breakpoint 3, 0x0001051c in main () at P3.c:7
7      for(i = 0; i < 5; i++)
(gdb) info register cpsr
cpsr      0x20000010      536870928
(gdb) disass
Dump of assembler code for function main:
0x00010498 <+0>:  push    {r11, lr}
0x0001049c <+4>:  add     r11, sp, #4
0x000104a0 <+8>:  sub     sp, sp, #32
0x000104a4 <+12>: ldr     r3, [pc, #200] ; 0x10574 <main+220>
0x000104a8 <+16>: ldr     r3, [r3]
0x000104ac <+20>: str     r3, [r11, #-8]
0x000104b0 <+24>: mov     r3, #0
0x000104b4 <+28>: ldr     r3, [pc, #188] ; 0x10578 <main+224>
0x000104b8 <+32>: sub     r12, r11, #28
0x000104bc <+36>: mov     lr, r3
0x000104c0 <+40>: ldm     lr!, {r0, r1, r2, r3}
0x000104c4 <+44>: stmia   r12!, {r0, r1, r2, r3}
0x000104c8 <+48>: ldr     r3, [lr]
0x000104cc <+52>: str     r3, [r12]
0x000104d0 <+56>: mov     r3, #0
0x000104d4 <+60>: str     r3, [r11, #-32] ; 0xfffffffffe0
0x000104d8 <+64>: b       0x10524 <main+140>
0x000104dc <+68>: ldr     r3, [r11, #-32] ; 0xfffffffffe0
0x000104e0 <+72>: lsl     r3, r3, #2
0x000104e4 <+76>: sub     r2, r11, #4
0x000104e8 <+80>: add     r3, r2, r3
0x000104ec <+84>: ldr     r2, [r3, #-24] ; 0xffffffffe8
0x000104f0 <+88>: ldr     r3, [r11, #-32] ; 0xfffffffffe0
0x000104f4 <+92>: sub     r3, r2, r3
0x000104f8 <+96>: mov     r0, r3
0x000104fc <+100>: bl      0x10ae4 <__floatsisf>
0x00010500 <+104>: mov     r3, r0
0x00010504 <+108>: mov     r1, r3
0x00010508 <+112>: ldr     r0, [r11, #-36] ; 0xffffffffdc
0x0001050c <+116>: bl      0x1094c <__aeabi_fadd>
0x00010510 <+120>: mov     r3, r0
0x00010514 <+124>: str     r3, [r11, #-36] ; 0xffffffffdc
=> 0x00010518 <+128>: ldr     r3, [r11, #-32] ; 0xfffffffffe0
0x0001051c <+132>: add     r3, r3, #1
0x00010520 <+136>: str     r3, [r11, #-32] ; 0xfffffffffe0
```

Q2)

```
0xff7bca40 in ?? ()
(gdb) b main
Breakpoint 1 at 0x104a4: file P3.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, main () at P3.c:3
3      int main() {
(gdb) n
4          int a[] = {1, 2, 3, 3, 5};
(gdb) n
7          for(i = 0; i < 5; i++)
(gdb) info register cpsr
cpsr      0x60000010      1610612752
(gdb) n
9              f = f + 1*(a[i] - i);
(gdb) info register cpsr
cpsr      0x80000010      -2147483632
(gdb)
```

Q3)

```
(gdb) n
9              f = f + 1*(a[i] - i);
1: i = 1
(gdb) n
7          for(i = 0; i < 5; i++)
1: i = 1
(gdb) n
9              f = f + 1*(a[i] - i);
1: i = 2
(gdb) n
7          for(i = 0; i < 5; i++)
1: i = 2
(gdb) n
9              f = f + 1*(a[i] - i);
1: i = 3
(gdb) n
7          for(i = 0; i < 5; i++)
1: i = 3
(gdb) n
9              f = f + 1*(a[i] - i);
1: i = 4
(gdb) n
7          for(i = 0; i < 5; i++)
1: i = 4
(gdb) info register cpsr
cpsr      0x20000010      536870928
(gdb) n
12         printf("float: %f\n", f);
1: i = 5
(gdb) info register cpsr
cpsr      0x20000010      536870928
```

Q1) CPSR value is 0x20000010,
which is 0010 0000 0000 0000 0000 0000 0001 0000
31st bit = N = 0
20th bit = Z = 0
29th bit = C = 1
28th bit = V = 0

Q2) CPSR value is 0x60000010,
which is 0110 0000 0000 0000 0000 0000 0001 0000
31st bit = N = 0
20th bit = Z = 1
29th bit = C = 1
28th bit = V = 0

Q3) CPSR value is 0x20000010,
which is 0010 0000 0000 0000 0000 0000 0001 0000
31st bit = N = 0
20th bit = Z = 0
29th bit = C = 1
28th bit = V = 0