```
int mod(int num, int den) {
   return num % den;
}
```

```
mod:
   push    {lr}
   sub     sp, sp, #12
   str     r0, [sp, #4]
   str     r1, [sp]
   ldr     r3, [sp, #4]
   ldr     r1, [sp]
   mov     r0, r3
   bl      __aeabi_idivmod
   mov     r3, r1
   mov     r0, r3
   add     sp, sp, #12
   pop     {pc}
```

```
isprime:
   push {lr}
   sub sp, #8
   str r0, [sp]
   mov r1, #1
   str r1, [sp, #4]
loop:
   ldr r1, [sp, #4]
   add r1, #1
   str r1, [sp, #4]
   ldr r0, [sp]
   cmp r1, r0
   bge prime
   bl mod
   cmp r0, #0
   beq notprime
   b loop
prime:
   mov r0, #1
   b end
notprime:
   mov r0, #0
end:
   add sp, #8
   pop {pc}
```

r0	15 2 15
r1	2 2 2 2 3
r2	???
r3	15 1
r13(sp)	0x120 0x11C 0x114 0x110
r14(lr)	0x100010 loop+28 mod+32
r15(pc)	isprime mod divmod

0x104 0x110 0x114

loop+28

0x0f4	
0x0f8	
0x0fc	
0x100	
0x104	2
0x108	15
0x10c	???
0x110	loop+28
0x114	15
0x118	2 2 3
0x11c	0x100010
0x120	

div mod

local vars

main

$ gcc -o isprime -fomit-frame-pointer ... 5

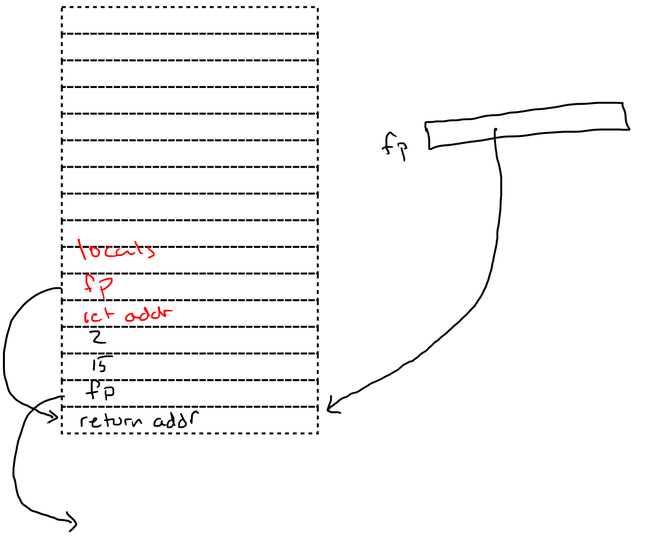
```
int mod(int num, int den) {
   return num % den;
}
```

```
$ gcc -o isprime isprime.o main.c
```

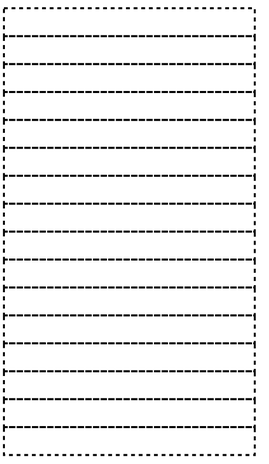
```
mod:
  push    {fp, lr}
  add     fp, sp, #4
  sub     sp, sp, #8
  str     r0, [fp, #-8]
  str     r1, [fp, #-12]
  ldr     r3, [fp, #-8]
  mov     r0, r3
  ldr     r1, [fp, #-12]
  bl      10690 <__aeabi_idivmod>
  mov     r3, r1
  mov     r0, r3
  sub     sp, fp, #4
  pop     {fp, pc}
```

fp = r11

Memory



Memory



Memory