```
                .file   "gcd.c"
                .text
.globl gcd
                .type   gcd, @function
gcd:
                pushl   %ebp
                movl    %esp, %ebp
                subl    $20, %esp
                jmp     .L2
.L3:
                movl    12(%ebp), %eax
                movl    %eax, 8(%ebp)       m=n
                movl    -4(%ebp), %eax
                movl    %eax, 12(%ebp)      n = r
.L2:
                movl    8(%ebp), %edx
                movl    %edx, %eax
                sarl    $31, %edx
                idivl   12(%ebp)            %edx = m % n,
                movl    %edx, -4(%ebp)      quotient in %eax,
                cmpl    $0, -4(%ebp)        remainder in %edx
                jne     .L3
                movl    12(%ebp), %eax      return n(caller expects
                leave                       value in %eax)
                ret
                .size   gcd, .-gcd
                .ident  "GCC: (Ubuntu 4.3.3-5ubuntu4) 4.3.3"
                .section .note.GNU-stack,"",@progbits
```

Assembler directives start with "."
"This will be executable code"
"gcd is a linker-visible label"

move %ebp to %esp and pop %ebp from the stack"

Pop a return address from the stack and branch to it