```
/*****************************************/
/* Introduction to Compiler Construction */
/*                                       */
/* Christoph Kirsch                      */
/* University of Salzburg                */
/*                                       */
/* encoding/decoding all binary formats  */
/*****************************************/

#include <stdlib.h>
#include <stdio.h>

int op;
int a;
int b;
int c;

int instruction;

encode() {
  // in compiler and linker!
  // assuming: 0 <= op <= 2^6-1 = 63
  // assuming: 0 <= a <= 2^5-1 = 31
  // assuming: 0 <= b <= 2^5-1 = 31
  // assuming: -32768 = -2^15 <= c <= 2^26-1 = 67108863
  // assuming: if c > 2^15-1 = 32767 then a == 0 and b == 0
  if (c < 0)
    c = c + 65536; // 0x10000: 2^16
  // if << is not available
  // replace (x << 5) by (x * 32) and (x << 16) by (x * 65536)
  instruction = (((((op << 5) + a) << 5) + b) << 16) + c;
}

decode() {
  // in linker and emulator!
  // works for format F1 and F2 but not F3
  // assuming: 0 <= instruction <= 2^32-1
  op = (instruction >> 26) & 63; // 0x3F: 6 lsbs
  a = (instruction >> 21) & 31; // 0x1F: 5 lsbs
  b = (instruction >> 16) & 31; // 0x1F: 5 lsbs
  c = instruction & 65535; // 0xFFFF: 16 lsbs
  if (c >= 32768)
    c = c - 65536; // 0x10000: 2^16
}

decodeF3() {
  // in linker and emulator!
  // works for format F3 only
  // assuming: 0 <= instruction <= 2^32-1
  op = (instruction >> 26) & 63; // 0x3F: 6 lsbs
  c = instruction & 67108863; // 0x3FFFFFF: 26 lsbs
}

main() {
  op = 63;
  a = 31;
  b = 31;
  c = -32768;

  encode();
  decode();

  printf("op: %d, a: %d, b: %d, c: %d (instruction: %d)\n", op, a, b, c,
instruction);
}
```