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

const char *unsigned_to_binary(unsigned long msg, int size) {
      char* b = (char *)malloc(size + 1);
      b[0] = '\0';
      int z;
      for (z = (1<<(size-1)); z > 0; z >>= 1) {
            strcat(b, ((msg & z) == z) ? "1" : "0");
      }
      return b;
}

//Defines a function called crc_remainder that recieves 4 arguments. It is used
calculate the remainder.
unsigned long crc_remainder(unsigned long msg, unsigned long check, int mlen, int
clen) {
      //Declares a long variable called newms. It is defined as the bits in msg
being left shifted by clen-1.
      unsigned long newmsg = msg << (clen-1);
      //Declares a new long variable called n.
      unsigned long n;
      //Declares a new integer variable called n.
      int i;
      //Instantiates a for-loop where the starting iterating value is equal to
mlen.
      //The value of i is reduced by 1 with each iteration until it falls below
zero.
      for (i = mlen; i > 0; i--) {
            if ((newmsg & (1 << (i+clen-2))) != 0) {
                  //Defines n as the bits in check being left shifted by i-1.
                  n = check << (i - 1);
                  //Redefines newmsg by performing newmsg XOR n.
                  newmsg = newmsg ^ n;
            }
      }
      //Returns final message that is evenly divisble by divisor polynomial. It is
the newmsg manipulated by this function.
      return newmsg & ((1 << clen) - 1);
}

int main() {
      unsigned long msg = 0xC74A;
      unsigned long divisor = 0xB;
      unsigned long newmsg = (msg << 3) + crc_remainder(msg, divisor, 16, 4);
      unsigned long rem, newrem;
      printf("Message is: ");
      printf("%s\n", unsigned_to_binary(msg, 16));
      printf("Divisor is: ");
      printf("%s\n", unsigned_to_binary(divisor, 4));
      rem = crc_remainder(msg, divisor, 16, 4);
      printf("Remainder of message and divisor: ");
      printf("%s\n", unsigned_to_binary(rem, 3));
      printf("Message with added bits is: ");
      printf("%s\n", unsigned_to_binary(newmsg, 19));
      newrem = crc_remainder(newmsg, divisor, 19, 4);
      printf("Remainder is: ");
      printf("%s\n", unsigned_to_binary(newrem, 3));
```

```
        printf("\nEXAMPLE FROM BOOK:\n\n");
        char * ptr;
        msg = strtoul("10011010", &ptr, 2);
        divisor = strtoul("1101", &ptr, 2);

        printf("Message is: ");
        printf("%s\n", unsigned_to_binary(msg, 8));
        printf("Divisor is: ");
        printf("%s\n", unsigned_to_binary(divisor, 4));
        rem = crc_remainder(msg, divisor, 8, 4);
        printf("Remainder of message and divisor: ");
        printf("%s\n", unsigned_to_binary(rem, 3));
        newmsg = (msg << 3) + rem;
        printf("Message with added bits is: ");
        printf("%s\n", unsigned_to_binary(newmsg, 11));
        newrem = crc_remainder(newmsg, divisor, 11, 4);
        printf("Remainder is: ");
        printf("%s\n", unsigned_to_binary(newrem, 3));
        return 0;
}
```