```c
/* File:     main.c
 * Purpose:  Use x86 assembly to compute the nth Fibonacci number
 *
 * Compile:  gcc -g -Wall -o fibo main.c fibo.s
 * Run:      ./fibo <n>
 *
 * Input:    none
 * Output:   The nth Fibonacci number computed using the recursive
 *           formula:
 *
 *                 F_0 = 0
 *                 F_1 = 1
 *                 F_n = F_n-1 + F_n-2, n >= 2
 *
 * Notes:
 * 1.  Compiling with ALL_IN_C defined will use a C function to
 *     compute the Fibonacci numbers.  For example,
 *
 *        gcc -g -Wall -o fiboc main.c -DALL_IN_C
 *        ./fiboc <n>
 *
 * 2.  This version should be compiled and run on a 64-bit Linux
 *     system.
 */
#include <stdio.h>
#include <stdlib.h>

long Fibo(long n);

int main(int argc, char* argv[]) {
   long n, result;

   if (argc != 2) {
      fprintf(stderr, "usage: %s <n>\n", argv[0]);
      exit(0);
   }

   n = strtol(argv[1], NULL, 10);
   if (n < 0) {
      fprintf(stderr, "n must be >= 0\n");
      exit(0);
   }

   result = Fibo(n);

   printf("F_%ld = %ld\n", n, result);

   return 0;
} /* main */

#ifdef ALL_IN_C
long Fibo(long n) {
   long f_n1, f_n2, f;

   if (n == 0)
      return 0;
   else if (n == 1)
      return 1;
   else {
      f_n1 = Fibo(n-1);
      f_n2 = Fibo(n-2);
      f = f_n1 + f_n2;
      return f;
   }
}  /* Fibo */
#endif
```