Goal: To determine the processing efficiency (in number of bits per cycle) of the "non-restoring" division technique with shift-over-0/1.

1. Write a program to simulate the process of an unsigned divider – a $2n$-bit numerator (N) and an n-bit divisor (D) leading to an $(n + 1)$-bit quotient (Q) and n-bit remainder (R) using the non-restoring technique with simple shift-over 0's and 1's using $\{-1/2, 1/2\}$ as comparators, $\{-D, D\}$ as subtractors.

2. Results required include:

   - program listing,

   - a plot showing the average number of bits "processed" per addition/subtraction-cycle (or the average number of quotient bits obtained per addition/subtraction process as defined in the class) from 10,000 simulation runs, for $n = 16, 18, \cdots, 32$, and

   - **using your program** to provide a demonstration of step-by-step (per subtraction-cycle) intermediate results for a 16-bit division with
     $N = 00001001101000001010101010101010$ and $D = 0111101100101010$,
     and clearly indicate what exactly transpires in each cycle.

   Note: In order to avoid overflow, you should always fix the first 2 bits of D (divisor) to 01 to start with, i.e. by assuming D is normalized already.