Definition 1. If \( f, g \in \mathbb{C}^G \), then define the convolution \( f \ast g \in \mathbb{C}^G \) by
\[
(f \ast g)(a) = \sum_{x \in G} f(x)g(a - x)
\]

Exercise 2. \( \hat{f} \ast \hat{g} = \hat{f} \cdot \hat{g} \)

Exercise 3. (a) Note that, in base ten arithmetic,
\[
12 = 3 \cdot 4 \quad \text{and} \quad 56 = 7 \cdot 8.
\]
Prove that these are the only solutions, in any base \( t \), to the equation
\[
a_1 a_2 \overset{t}{=} a_3 a_4,
\]
where \( a_1, a_2, a_3, a_4 \) are an increasing sequence of consecutive one-digit integers, i.e., if the base is \( t \) then \( 0 \leq a_i \leq t - 1 \), \( a_i - a_{i-1} = 1 \) and the meaning of \( \overline{a_1 a_2} \overset{t}{=} \) is \( a_1 t + a_2 \). So \( t = 10 \) is the only base where the equation is solvable!

(b) Prove that if \( a_1, a_2, a_3, a_4 \) are consecutive and \textit{decreasing}, then there are two solutions to
\[
a_1 a_2 = a_3 a_4 \overset{t}{=}
\]
when \( t = 6 \), and no solutions in any other base.

Exercise 4. Find two infinite sets \( A, B \) of non-negative integers such that every non-negative integer can be represented uniquely as a sum \( a + b \), where \( a \in A, b \in B \).

If you can do this, now characterize all such pairs of subsets \( A, B \).

The Karatsuba-Ofman Algorithm.

The Karatsuba-Ofman algorithm provides a striking example of how the “Divide and Conquer” technique can achieve an asymptotic speedup over an ancient algorithm.

The classroom method of multiplying two \( n \)-digit integers requires \( \Theta(n^2) \) digit operations. We shall show that a simple recursive algorithm solves the problem in \( \Theta(n^\alpha) \) digit operations (addition, multiplication, bookkeeping (such as copying digits and maintaining links)), where \( \alpha = \log_2 3 \approx 1.58 \). This is a considerable improvement of the asymptotic order of magnitude of the number of digit-operations.

Note: The Fourier transform-based algorithm of Schönhage and Strassen further improves this (to \( O(n \log n) \)), but the Karatsuba-Ofman algorithm is far more elementary.
We describe the procedure in pseudocode.

**Procedure** KO($X,Y$)

**Input:** $X,Y$: $n$-digit integers.

**Output:** the product $X \times Y$.

**Comment:** We assume $n = 2^k$, by prefixing $X,Y$ with zeros if necessary.

1. if $n = 1$ then use multiplication table to find $T := X \times Y$
2. else split $X,Y$ in half:
   3. $X := 10^{n/2}X_1 + X_2$
   4. $Y := 10^{n/2}Y_1 + Y_2$
   5. Comment: $X_1, X_2, Y_1, Y_2$ each have $n/2$ digits.
   6. $U := KO(X_1, Y_1)$
   7. $V := KO(X_2, Y_2)$
   8. $W := KO(X_1 - X_2, Y_1 - Y_2)$
10. $T := 10^n U + 10^{n/2} Z + V$
11. Comment: So $U = X_1 \times Y_1$, $V = X_2 \times Y_2$, $W = (X_1 - X_2) \times (Y_1 - Y_2)$, and therefore $Z = X_1 \times Y_2 + X_2 \times Y_1$. Finally we conclude that $T = 10^n X_1 \times Y_1 + 10^{n/2} (X_1 \times Y_2 + X_2 \times Y_1) + X_2 \times Y_2 = X \times Y$.
12. return $T$

**Note:** This is a recursive algorithm: during execution, it calls smaller instances of itself.

Let $M(n)$ denote the number of digit-multiplications (line 1) required by the Karatsuba–Ofman algorithm when multiplying two $n$-digit integers ($n = 2^k$). In lines 6,7,8 the procedure calls itself three times on $n/2$-digit integers; therefore

$$M(n) = 3M(n/2), \quad M(1) = 1.$$  \hspace{1cm} (1)

**Exercise 5.** Prove: $M(n) = n^\alpha$, where $\alpha = \log_2 3$ (assuming $n = 2^k$).

It would seem that we reduced the number of digit-multiplications to $n^{\log_2 3}$ at the cost of an increased number of additions (lines 9, 10). Appearances are deceptive: actually, the procedure achieves similar savings in terms of the total number of digit-operations (additions and bookkeeping as well as multiplications).

To see this, let $T(n)$ be the total number of digit-operations (additions, multiplications and bookkeeping) required by the Karatsuba–Ofman algorithm. Then

$$T(n) = 3T(n/2) + O(n)$$  \hspace{1cm} (2)

where the term $3T(n/2)$ comes, as before, from lines 6,7,8; the additional $O(n)$ term is the number of digit-additions required to perform the additions and subtractions in lines 9 and 10. The $O(n)$ term also includes bookkeeping costs.

**Exercise 6.** Prove: Equation (2) implies $T(n) = O(n^\alpha)$ where $\alpha = \log_2 3$. 