Universidade Federal do Rio Grande do Norte
Instituto Metrpole Digital

Programming Language I • IMD0030

◁ Exercises on Recursion ▷

March 19, 2018

1. Spherical objects, such as cannonballs, can be stacked to form a pyramid with one cannonball at the top, sitting on top of a square composed of four cannonballs, sitting on top of a square composed of nine cannonballs, and so forth, as shown in Figure 1. Write a recursive function `cannonball` that takes as its argument the height of the pyramid and returns the number of cannonballs it contains. Your function must operate recursively and must not use any iterative constructs, such as `while` or `for`. Also, write a program, `cannonballs.cpp` to test your function. Your program should receive and validate the height of the pyramid through command line arguments, and print the total number of cannonbals that can be stacked to form a pyramid of the requested height.

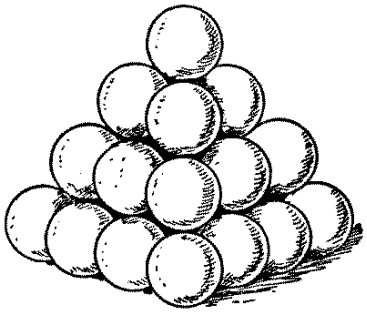
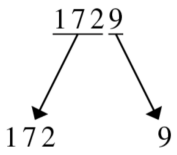


Figure 1: A stack of cannonballs with base=4.

2. Unlike many programming languages, C++ does not include a predefined operator that raises a number to a power. As a partial remedy for this deficiency, write a recursive implementation of a function `int raiseToPower(int n, int k)` that calculates n^k. The recursive insight that you need to solve this problem is the mathematical property that

$$n^k = \begin{cases} 1 & \text{if } k \text{ is 0} \\ n \times n^{k-1} & \text{otherwise} \end{cases}$$

3. Write a recursive function `digitSum(n)` that takes a nonnegative integer and returns the sum of its digits. For example, calling `digitSum(1729)` should return $1 + 7 + 2 + 9$, which is 19.



Each of the resulting integers is strictly smaller than the original and thus represents a simpler case.

4. The **greatest common divisor** (often abbreviated to **gcd**) of two nonnegative integers is the largest integer that divides evenly into both. In the third century BCE, the Greek mathematician Euclid discovered that the greatest common divisor of x and y can always be computed as follows:

   - If x is evenly divisible by y, then y is the greatest common divisor.
   - Otherwise, the greatest common divisor of x and y is always equal to the greatest common divisor of y and the remainder of x divided by y.

   Use Euclid's insight to write a recursive function `gcd(x, y)` that computes the greatest common divisor of x and y.

5. The **digital root** of an integer n is defined as the result of summing the digits repeatedly until only a single digit remains. For example, the digital root of 1729 can be calculated using the following steps:

$$\begin{array}{lrcl} \text{Step 1:} & 1+7+2+9 & \rightarrow & 19 \\ \text{Step 2:} & 1+9 & \rightarrow & 10 \\ \text{Step 3:} & 1+0 & \rightarrow & 1 \end{array}$$

   Because the total at the end of step 3 is the single digit 1, that value is the digital root.

   Write a function `digitalRoot(n)` that returns the digital root of its argument. Although it is easy to implement `digitalRoot` using the `digitSum` function from exercise 3 and a `while` loop, part of the challenge of this problem is to write the function recursively without using any explicit loop constructs.

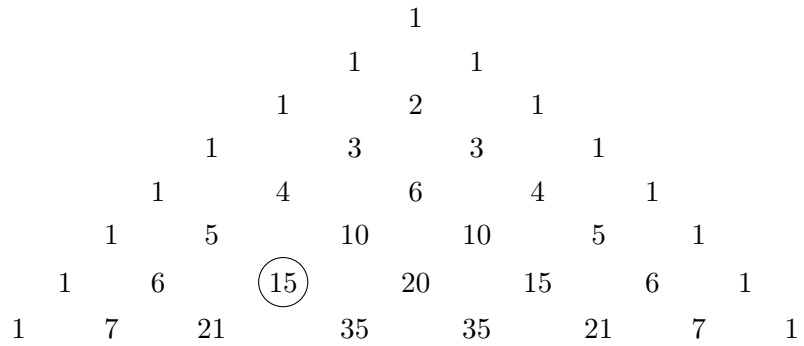
6. The mathematical combinations function $c(n, k)$ is usually defined in terms of factorials, as follows:

$$c(n,k) = \frac{n!}{k! \times (n-k)!}$$

   The values of $c(n, k)$ can also be arranged geometrically to form a triangle in which n increases as you move down the triangle and k increases as you move from left to right. The resulting structure, which is called **Pascal's Triangle** after the French mathematician Blaise Pascal, is arranged like this:

$$\begin{array}{ccccccccc} & & & & c(0,0) & & & & \\ & & & c(1,0) & & c(1,1) & & & \\ & & c(2,0) & & c(2,1) & & c(2,2) & & \\ & c(3,0) & & c(3,1) & & c(3,2) & & c(3,3) & \\ c(4,0) & & c(4,1) & & c(4,2) & & c(4,3) & & c(4,4) \end{array}$$

Pascal's Triangle has the interesting property that every entry is the sum of the two entries above it, except along the left and right edges, where the values are always 1. Consider, for example, the circled entry in the following display of Pascal's Triangle:



This entry, which corresponds to $c(6, 2)$, is the sum of the two entries—5 and 10—that appear above it to either side. Use this relationship between entries in Pascal's Triangle to write a recursive implementation of the $c(n, k)$ function that uses no loops, no multiplication, and no calls to `fact`.

Also, write a program, `pascal.cpp` to test your function. Your program should receive and validate the height of the Pascal's Triangle through command line arguments, and print the complete triangle.

7. Write a recursive function called `unique` that eliminates repetition of element from the range [`first`, `last`), and returns a past-the-end pointer for the new logical end of the range. Removing is done by shifting the elements in the range in such a way that elements to be erased are overwritten. Relative order of the elements that remain is preserved and the physical size of the container is unchanged. For example:

$$[\ 1\ \ 5\ \ 8\ \ 1\ \ 5\ \ 2\ \ 3\ \ 3\ \ ] \text{ becomes } [\ 1\ \ 5\ \ 8\ \ 2\ \ 3\ ].$$

Hint: It might be helpfull to write an auxiliary recursive function `find` that returns `true` if it finds a target value within a range [`first`, `last`).

8. A *palindrome* is a string that reads identicaly backward and forward, such as "`level`" and "`noon`". Write a recursive function called `palindrome` that receives a string and returns `true` if the string is a palindrome, or `false` otherwise.

   The insight you need to design a recursive solution is that any palindrome longer than a single character must contain a shorter palindrome in its interior. For example, the string "`level`" consists of the palindrome "`eve`" with "`l`" at each end.

∽ FIM ∽