

Lab Assignment 1: Introduction

(Textbook Chapter 1)

(Exercise 3/p. 23)

(Exercise 4/p. 23) **Solution:** A solution to take care of a single leading zero would replace Step 8 with:

1. if ($c_m = 0$) then
2. print out $c_{m-1}c_{m-2}\dots c_0$
3. else
4. print out $c_m c_{m-1}\dots c_0$

A solution to take care of multiple leading zeros would replace Step 8 with:

1. $i = m$
2. while ($c_i = 0$ AND $i \geq 0$) do
3. $i = i - 1$
4. if $i \geq 0$ then
5. print out $c_i c_{i-1}\dots c_0$

(Exercise 6/p. 23) **Solution:** The algorithm in Figure 1.3(a) is a better general purpose algorithm since it can easily be rewritten to allow the user to wash their hair any number of times. In fact, the number of washings could be a user input. The algorithm in Figure 1.3(b) can be extended to more washings only by writing two more lines of code for each washing, and it can't handle arbitrary user input specifying the number of washings.

(Exercise 7/p. 23) **Solution:** (b), No, the algorithm does not work correctly when the inputs are 0 and 32, because in Step 2 it tries to divide 32 by 0, an operation that is undefined. We can correct this by rewriting as follows:

1. Get two positive integers. Call larger one I and smaller one J .
2. If $J = 0$ then
3. print out "ERROR: division by zero not allowed".
4. else
5. $R = \text{the remainder of } I/J$
6. while (R is not equal to 0) do
7. $I = J$
8. $J = R$
9. $R = \text{the remainder of } I/J$
10. Print out the answer

(Exercise 8/p. 24) **Solution:** There are $25!$ possible paths. This is about $1.5 \cdot 10^{25}$ paths. If the computer can analyse 10^7 paths per second, it will take $1.5 \cdot 10^{25}/10^7 = 1.5 \cdot 10^{18}$ seconds to analyze all paths. This is about a trillion years; clearly not feasible..A feasible alternative that does not necessarily yield the *best* answer is to start at an arbitrary city and keep visiting the still-unvisited city that is closest to the one you are currently in. This can be done by looking at distances from the city we are currently in to all still-unvisited cities and picking the shortest one to visit next. This is unlikely the best route, but it's probably much better than choosing a route randomly, and it can be done efficiently.

(Exercise 9/p. 24) **Solution:**

1. Get numbers a and b
2. $c = 0$
3. $i = 1$
4. while $i \leq b$ do
5. $c = c + a$
6. $i = i + 1$
7. Print out the answer c