MOST 2012-2013 - Algorithmics - Lab sessions

Exercise 1.1 : Alphabetical order

Write an algorithm that asks 10 words to the user and display them in alphabetical order.

Exercise 1.2 : Unscrabble

Write an algorithm that asks a word to the user and return all the letters of the word in alphabetical order. For example, word "algorithmics" will be returned as "acghiilmorst".

Exercise 1.3 : Palindrome

A palindrome is a word that may be read the same way in either direction. For example, "radar". Write an algorithm that asks a word to the user and tells him if it is a palindrome or not.

Exercise 1.4 : Factorial

Write an algorithm computing the factorial of a number.

Exercise 1.5 : Prime number

A prime number is a number that can only be divided by 1 and by itself. Write an algorithm that asks for a number and tells to the user if this number is prime or not.

Exercise 1.6 : Leap year

A leap year (or bissextile year) is a year containing one additional day. Since 1582, in the Gregorian calendar, a year is a leap year if it is a multiple of 4, except when the year is a multiple of 100 that is not a multiple of 400. Write an algorithm that asks a year to the user and tells him if it is bissextile or not.

Exercise 1.7 : Tomorrow is another day

Write an algorithm that asks for a date with format "YYYY/MM/DD" and gives the day after this date. For example, if the user enters "2012/12/31", you have to return "2013/01/01". Take care to the number of days in a month, and to leap years...

Exercise 1.8 : Binary encoding

Numbers are written in binary numeral system (with only 0's and 1's) in a computer. To convert a number for decimal numeral system to binary numeral system, the number is divided by 2 and the remainder is the least-significant bit. The result is again divided by two and its remainder is the next least-significant bit. This process repeats until the

quotient becomes zero. For example, 33 is written 100001 in binary. Write an algorithm that asks for a number and writes it in binary numeral system.

Exercise 1.9 : Roman encoding

Roman numerals are based on seven symbols :

Symbol	Value
I	1
V	5
X	10
L	50
C	100
D	500
M	1000

To write an arabic numeral as a roman one, the following rules have to be respected :

- A number written in Arabic numerals can be broken into digits. For example, 1903 is composed of 1, 9, 0, and 3. To write the Roman numeral, each of the non-zero digits should be treated separately. In the above example, 1,000 = M, 900 = CM, and 3 = III. Therefore, 1903 = MCMIII
- The symbols "I", "X", "C", and "M" can be repeated three times in succession, but no more. (They may appear more than three times if they appear non-sequentially, such as XXXIX.) "D", "L", and "V" can never be repeated.
- "I" can be subtracted from "V" and "X" only. "X" can be subtracted from "L" and "C" only. "C" can be subtracted from "D" and "M" only. "V", "L", and "D" can never be subtracted
- Only one small-value symbol may be subtracted from any large-value symbol.

Write an algorithm that asks for a number between 1 and 3999 and returns the corresponding roman numeral.

Exercise 2.1 : Accepted...or not ?

Required : v, an array of integers
Write an algorithm that computes the mean value of the values of v. Use the function Math.floor(mean) to round off this mean. Print a message depending on the value obtained :

- $[0-5[$ failure
- $[5-10[$ remedial lessons
- $[10-15[$ accepted
- $[15-20[$ accepted with honor

Exercise 2.2 : Magic square

Require : s, a square array of size n by n
Write an algorithm that checks if s is a magic square. A square table is magic if the following terms are equal :

- the sum of any column
- the sum of any row
- the sum of term in any of the 2 diagonal

Exercise 2.3 : Negative values

Require : v, an array of integers
Write an algorithm which finds the negative values of v. Write these values and give the number of negative values.

Exercise 2.4 : Closiest value

Require : v, an array of integers, i an integer
Write an algorithm that finds the element of v which is the closest to i (using the absolute difference).

Exercise 2.5 : Again and again and...

Require : v, an array of integers
Write an algorithm that gives the element of v which occurs the least, and its number of occurrences. Change the algorithm to find the element of v which occurs the most, and its number of occurrences.

Exercise 2.6 : Consecutive values

Require : v, an array of integers
Write an algorithm which gives the longest serie of consecutive values in v, and its length. For example, if the array is $(3, 4, 2, 3, 5, 6, 7, 13)$, the longest serie of consecutive values is $(5, 6, 7)$.
Change the algorithm to find the shortest serie of consecutive values in v, and its length. In previous example, the shortest serie is (13).

Exercise 2.7 : Sum of consecutive values

Require : v, an array of integers, i an integer
Write an algorithm which gives the series of i consecutive values in v which have the highest sum, and the value of this sum.
Change the algorithm to find the series of i consecutive values in v which have the lowest sum, and the value of this sum.

Exercise 3.1 : Fibonacci sequence

Require : i an integer
In mathematics, the Fibonacci sequence is defined as follows :
$F(0) = 1$
$F(1) = 1$
$F(n + 2) = F(n + 1) + F(n)$
Thus, the Fibonacci sequence starts as follows : 1, 1, 2, 3, 5, 8, 13, 21, ... Write an algorithm that computes F(i).

Exercise 3.2 : Factorial

Require : i an integer
Write a recursive algorithm that computes the factorial of i.

Exercise 3.3 : Dichotomic search

Require : v an array of sorted integers, i an integer
Write an algorithm, using a dichotomic search, that finds the element of v which is the closest to i. In computer science, a dichotomic search is a search algorithm that operates by spliting the search space in two distinct areas at each step. For instance, it is possible to split v in two areas. The search will go on with the area whose lower bound (respectively upper bound) is lower (respectively greater) than i.

Exercise 3.4 : Selection sort

Write an algorithm that asks for an array of numbers and sort them from smallest to largest by using the following method :

1. Find the minimum value in the list
2. Swap it with the value in the first position
3. Repeat the steps above for the remainder of the list (starting at the second position and advancing each time)

Exercise 3.5 : Bubble sort

Write an algorithm that asks for an array of numbers and sort them from smallest to largest by using the following method :

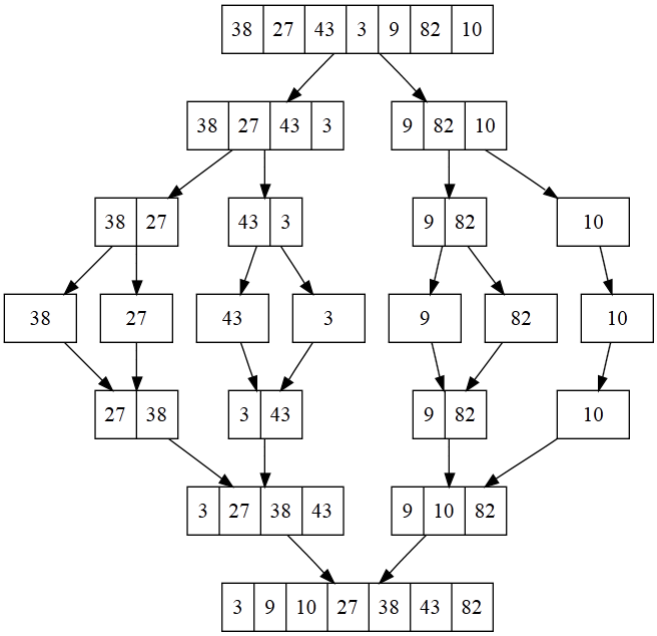
- Compare each pair of adjacent items and swap them if they are in the wrong order.
- The pass through the list is repeated until no swaps are needed, which indicates that the list is sorted.

Exercise 3.6 : Merge sort

Write an algorithm that asks for an array of numbers and sort them from smallest to largest by using the following method :

1. Divide the unsorted array into n array, each containing 1 element (an array of 1 element is considered sorted).
2. Repeatedly merge arrays to produce new arrays until there is only 1 array remaining. This will be the sorted array.

An example of this algorithm is presented on the following figure :



(Hint : Think on how to define a function that merges two sorted arrays and produces a merged array.)