UNIVERSIDAD TECNICA
FEDERICO SANTA MARIA

FÍSICA COMPUTACIONAL

Homework #2

Due to October 30

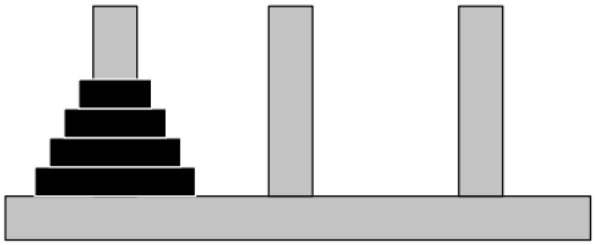
Please solve the following problems and email your solutions to both Prof. Florez and Prof. Carquín. Comprehensive instructions to handle your software are expected as well as a detailed description of your programs' structure.

Juan Manuel Florez y Edson Carquín
juanmanuel.florez@usm.cl; edson.carquin@usm.cl

Table of Contents

Problem #1:

a) Please create an algorithm that solves the ancient Towers of Hanoi puzzle. To solve the puzzle, you must come up with a series of steps to move a stack of different sized rings from one pole to another. They must be moved one at a time, using only simple intermediate pole, so that not ring is ever placed on top of a smaller ring.
b) You should provide us with an estimate of the O(n) asymptotic behavior for such an algorithm.
c) Please create a program in C++, that implements your algorithm in (a).
d) Be free to represent the output of your program by using a plot, a table or whatever you think is the best way to do it so. You could use an alternative language like Mathematica or Python in order to display the results obtained with your program, but in that case, you should provide us with precise instructions to manage properly your codes.



Problem #2:

a) Please create a C++ program that implement the Strassen's algorithm for matrix multiplication.
b) You should demonstrate that your implementation fulfills the O(n) for such an algorithm by using enough examples of variety n/2 x n/2 matrices. Those matrices should be fed to the program by using input-output instructions for plane files.
c) Use your program to try to determine, up to the hardware and resources capabilities you have, what would be the best range of matrices' sizes for the Strassen algorithm to be really useful in terms of the comparison with conventional square matrices multiplication. If with your resources you do not see any difference in realistic time, describe properly your computational experiment to reach that conclusion.

Problem #3:

Write a program using C++, which implements an algorithm to calculate recursively $\sqrt{2}$, with a precision of 6 decimal digits.

Problem #4:

Write a C++ program that reads a word from the keyboard, stores it in a string and checks whether the word is a palindrome. A palindrome reads the same from left to right as from right to left (otto, level and deed are examples of palindromes). Use the subscript operator [ ]. Modify the code to continually read and check new words and store the output into a file with the test word and the result of the test in the same line.

Problem #5:

a) Explain each of the following definitions. Indicate whether any are illegal and if so why.

```
(a) int* ip;
(b) string s, *sp = 0;
(c) int i; double* dp = &i;
(d) int* ip, ip2;
(e) const int i = 0, *p = i;
(f) string *p = NULL;
```

b) Given a pointer, p, can you determine whether p points to a valid object? if so, how? It not, why not?

c) Why is the first pointer initialization legal and the second illegal?

```
int i = 42;
void *p = &i;
long *lp = &i;
```