**ECE 1320 Optimization Methods**
**Winter 2003**

**Homework 4: Due in class Thursday February 4 2003**

- This test contains 6 problems. They allow you to earn 100 points.
- Show your work, as partial credit can be given. You will be graded not only on the correctness of your answer, but also on the clarity with which you express it. **Be neat**.
- **No late submissions will be accepted.**
- Only homework returned in a 9in $\times$ 12in envelope will be accepted. (If you cannot find such envelope, ask the Instructor.) Please, write your name and the class name (ECE 1320) on the envelope (write clearly, please).
- For the six problems an e-mail to the TA should be sent that contains the code and the executable of a program that implements the solutions to the problems as functions.

Write your name here: ____________________________________________

- **Problem # 1 [15 points]**. **(a)** Write a C++ Boolean function that given as input an array $A$ of integers and an integer $i$ determines whether $i$ appears in $A$ *exactly* twice. Determine an upper bound and a lower bound for this problem. **(b)** Do the same for the problem to determine whether $i$ appears *at least* twice in $A$.

- **Problem # 2 [15 points]**. Write a recursive C++ function that given an array of $n$ characters $A$ determines if the corresponding string is a *palindrome*. A palindrome is a word, verse or sentence (as in "Able was I ere I saw Elba") or a number (e.g., 1881) that reads the same backward or forward.

- **Problem # 3 [20 points]**. Write three C++ functions that given as input two integers $a > 0$ and $n = 2^k$, $k > 0$, compute $a^n$ in linear time, in logarithmic time and in constant time, respectively. (You can use C++ library functions from `math.h` such as `log()` and `exp()`.)

- **Problem # 4 [20 points]**. **(a)** Consider the following polynomial

$$P(x) = \sum_{k=0}^{n} a_k x^k.$$

Write a C++ function that given the array $A$ of the $n+1$ coefficients $a_0, a_1, \ldots, a_n$, $a_i > 0$, $0 \leq i \leq n$, and a value $x$, evaluates (i.e., returns) $P(x)$. Describe the worst-case time complexity of your solution.

**(b)** Consider the *Horner's rule* for evaluating $P(x)$.

$$P(x) = \sum_{k=0}^{n} a_k x^k = a_0 + x(a_1 + x(a_2 + \cdots + x(a_{n-1} + x a_n) \cdots)).$$

Write a C++ function that implements this rule, and determine its time complexity.

- **Problem # 5 [20 points]**. The recursive definition of the binomial coefficient when $k \leq n$

$$\binom{n}{k} = \begin{cases} 1 & \text{if } k = 0 \text{ or } k = n \\ \binom{n-1}{k-1} + \binom{n-1}{k} & \text{if } 0 < k < n. \end{cases}$$

suggests the definition of a (C++) function that returns $\binom{n}{k}$ which is different from the more "natural" recursive implementation. The binomial coefficient $\binom{n}{k}$ can be seen as the last entry $c_{n,k}$ of a $(n+1) \times (k+1)$ matrix which can be filled out line by line in the following way: $c_{0,j} = 0$, $1 \leq j \leq k$, $c_{i,0} = 1$, $0 \leq j \leq n$, and

$$c_{i,j} = c_{i-1,j-1} + c_{i-1,j}, 1 \leq i \leq n, 1 \leq j \leq k.$$

Use this characterization (also known as *Pascal's triangle*) to write a C++ function that taken as input $n$ and $k$ returns $\binom{n}{k}$. Discuss the time *and* the *space* complexity of your solution. Can this problem be solved in linear space?

- **Problem # 6 [20 points]**. (*First $n$ integers.*) Write an optimal, recursive C++ function that given as input an integer $n \geq 1$ construct two lists $\ell$ and $m$ such that $\ell = (1, 2, \ldots, n-1, n)$ and $m = (n, n-1, \ldots, 2, 1)$.