ECE 1320 Optimization Methods
Winter 2003

Homework 3: Due in class Thursday January 30 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** **[20 points]**. Write a C++ function `minMax( int n, int &min, int &max, int A[] )` that determines the minimum *and* the maximum elements of an array A of n integers and minimize the number of comparisons.

- **Problem # 2** **[20 points]**. Write an *optimal* Boolean C++ function that, given an array of n integers ≥ 0, returns *true* is there are two elements in the array whose sum equals 23, *false* otherwise.

- **Problem # 3** **[20 points]**. Write an *optimal* C++ function that, given an integer $n \geq 1$, prints out all powers of 7 up to n.

- **Problem # 4** **[10 points]**. Write two recursive C++ functions to compute the maximum among the n elements of an (unsorted) array of integers. One function should implement the search by scanning (recursively) the array. The other function should perform the search according to a "balanced" divide and conquer technique, i.e., at any recursive call the search range in the array should be halved.

- **Problem # 5** **[15 points]**. Consider the problem of calculating the binomial coefficient when $k \leq n$:

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

  Write a function that computes $\binom{n}{k}$ by implementing directly the given definition. Discuss the time complexity of your solution.

- **Problem # 6** **[15 points]**. Write a recursive C++ function that given as input an array A of n integers and an integer i counts how many times i appears in A.