Problem 1. A random variable is a numerical measurement of the outcome of a random process. Formally, it is a $real - valued$ function $X : \Omega \rightarrow \mathbb{R}$ defined on the sample space Ω - the input $i \in \Omega$ to X is an outcome of the random process, and the output $X(i)$ is a numerical value.

For this question let us define a random variable that is the sum of two dice. This random variable maps outcomes (pairs(i, j) with $i, j \in \{1, 2, \ldots, 6\}$) to real numbers according to the map, $(i, j) \rightarrow i + j$.

1. What is the expected value of the sum of two dice? Use brute force approach to find this value.
2. Use a more efficient way to find this expected value. (*Hint*: linearity of expectation).

Show your work.

Problem 2. Let M be an $n \times n$ matrix containing n^2 distinct integers in which the integers of each row are sorted in an increasing order (from left to right) and the integers of each column are sorted in an increasing order (from top to bottom).

1. Let x be one of the numbers in M. Write pseudo-code for an efficient algorithm to find the position of x in M.
2. What is the worst case number of comparisons made by your algorithm in *part* 1 as a function of n? Consider the following two cases:
   (a) Single comparison to reveal whether one of the numbers is greater than, less than, or equal to the other number.
   (b) Two comparisons to reveal whether one of the numbers is greater than, less than, or equal to the other number.

Problem 3. In this problem we want to find the smallest and the largest values in an array of n values in linear time. We can do this by first finding the largest value in $n - 1$ comparisons and then the smallest value in separately, in $n - 2$ comparisons, for a total of $2n - 3$ comparisons. Write pseudo-code for a linear runtime algorithm to find the largest and the smallest value in an array such that the total number of comparisons is $\frac{3n}{2} - 2$.

Problem 4 (Challenge problem not graded) Let A be an arbitrary (un-sorted) array of distinct (very large) positive integers. Describe a strategy for an efficient algorithm that finds the two largest and the two smallest numbers in A with an exact worst-case number of comparisons to be, $\frac{3n}{2} + 2log_2 n - 4$.