**TASK 1: What is the expected running time of the following C# code? Explain why.**

**Assume the array's size is n.**

**----------------------------------------------------------------------**

**long Compute(int[] arr)**

**{**

**long count = 0;**

**for (int i=0; i<arr.Length; i++)**

**{**

**int start = 0, end = arr.Length-1;**

**while (start < end)**

**if (arr[start] < arr[end])**

**{ start++; count++; }**

**else**

**end--;**

**}**

**return count;**

**}**

**----------------------------------------------------------------------**

Answer: Expected to run in quadratic timeO(n2), coz we have 2 nested loops (first is the for, second is the while).

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**TASK 2: What is the expected running time of the following C# code? Explain why.**

**Assume the input matrix has size of n \* m.**

**----------------------------------------------------------------------**

**long CalcCount(int[,] matrix)**

**{**

**long count = 0;**

**for (int row=0; row<matrix.GetLength(0); row++)**

**if (****matrix[row, 0] % 2 == 0)**

**for (int col=0; col<matrix.GetLength(1); col++)**

**if (matrix[row,col] > 0)**

**count++;**

**return count;**

**}**

**----------------------------------------------------------------------**

Answer: Expected to run in quadratic time O(n\*m) (full O(n\*(m-z)), where z is **matrix[row, 0] % 2 != 0**), coz we have 2 nested loops (first is the for n, second is for m).

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**TASK 3\*: What is the expected running time of the following C# code? Explain why.**

**Assume the input matrix has size of n \* m.**

**----------------------------------------------------------------------**

**long CalcSum(int[,] matrix, int row)**

**{**

**long sum = 0;**

**for (int col = 0; col < matrix.GetLength(0); col++)**

**sum += matrix[row, col];**

**if (row + 1 < matrix.GetLength(1))**

**sum +=** **CalcSum(matrix, row + 1);**

**return sum;**

**}**

**Console.WriteLine(CalcSum(matrix, 0));**

**----------------------------------------------------------------------**

Answer: Expected to run in quadratic time O(n\*m), coz we call a new column loop for every new row, which we call recursively in “**CalcSum(matrix, row + 1);**”. Basicly two nested loops one for column and second for row.