**CS2102 - Data Structures**
**Project 2**

In Lecture 3, you have learned how to measure the time needed by a certain process or block of code experimentally and analytically.
Using the Dynamic array example, two different implementations for expandStorage() method were shown as follows:

Implementation 1:

```
private void expandStorage()
{
    int[] newStorage = new int[_storage.Length + 1];
    for(int i = 0; i < _storage.Length; i++)
        newStorage[i] = _storage[i];
}
```

Implementation 2:

```
private void expandStorage()
{
    int[] newStorage = new int[_storage.Length * 2];
    for(int i = 0; i < _storage.Length; i++)
        newStorage[i] = _storage[i];
}
```

The testing method was as follows:

```
static void arrayListTest()
{
    DynamicArray customArray = new DynamicArray(1);
    customArray.add(4);
    customArray.add(5);
    customArray.add(6);
    customArray.print();

    try
    {
        customArray[3] = 20;
    }
    catch (Exception ex)
    {
        Console.WriteLine(ex.Message);
    }
    customArray.add(20);
    customArray.print();
    customArray.insert(2, 200);
    customArray.print();
}
```

Write a similar method to test the two different implementations for expandStorage() method using different data sizes starting from 10000 up to 200000 with increment of 10000.

Hint:

```
var file= File.Create("./result.txt");
StreamWriter writer = new StreamWriter(file);
for (int size = 10000; size <= 200000; size = size + 10000)
{
     ……..
     writer.WriteLine(size + "\t" + sw.ElapsedMilliseconds.ToString() )
}
```

A sample of your result should be as follows:
10000 152
20000 616
30000 1372

Using Microsoft Excel, draw the relation between the numbers of elements and the execution time in milliseconds for the two different implementations.