BİL 212
Lab 2

1. Given an absolute path (starting with a '/' character) of a file or a directory in a Unix-like system, find the simplest canonical path to the same file or directory. (The canonical path refers to the shortest absolute path in this context which does not include any '.' or '..'.)

   '.' represents the current directory
   '..' represents the parent directory, the root directory does not have a parent directory which means '/..' is equivalent to '/'.
   Multiple occurrences of '/' such as '///' is equivalent to '/'.

   In the output, the path should not contain '.' or '..'. It also should not end with a trailing '/'.
   You can assume the given path is a valid Unix path.

   Example:
   Input: "/a/b/../c/./d/../file.txt"
   Ouptut: "/a/c/file.txt"

2. Given an encoded message you are asked to decode the message. The encoding format is k[encoded_string] where k is the number of times the encoded string is repeated. For example, 3[x] means "xxx" and 2[a2[cd]] means "acdcdacdcd".

   The encoding format restricts the values of k so that $0 < k < 10$.
   Messages consist of numbers and letters.
   Numbers are only used as the repetition count.

   Example:
   Input: 1[abc2[de]]
   Output: abcdede

3. Perform an experimental analysis that compares the relative running times of the methods given below. For your analysis to be meaningful you should try running these methods for different values of N such as 20000, 40000, 80000. Now, do a theoretical analysis, and see if your experimental analysis agrees with your theoretical analysis.
   (N = size of the input array)

```
public static int[] method1(int[] array)
{
	int k = 10;
	for (int i = k/2 ; i < array.length - k/2 ; i++)
	{
		for (int j = i - k/2; j < i + k/2 ; j++)
		{
			array[i] += array[j];
		}
		array[i] /= k;
	}
	return array;
}

public static int[] method2(int[] array)
{
	int k = array.length/10;
	for (int i = k/2 ; i < array.length - k/2 ; i++)
	{
		for (int j = i - k/2; j < i + k/2 ; j++)
		{
			array[i] += array[j];
		}
		array[i] /= k;
	}
	return array;
}
public static int[] method3(int[] array)
{
	int k = array.length%10 + 1;
	for (int i = k/2 ; i < array.length - k/2 ; i++)
	{
		for (int j = i - k/2; j < i + k/2 ; j++)
		{
			array[i] += array[j];
		}
		array[i] /= k;
	}
	return array;
}
```