F28DA1 **Tutorial Sheet 1**
Data Structures and Algorithms

Compression

1. Suppose you have analysed a file and determined that the frequency of occurrence of certain characters is as follows:

Character	a	b	c	d	e	f
Occurences	15	7	5	8	30	10

a) Construct the Huffman tree for the characters
b) List the codes for each character
c) Use the tree to compress the following strings:
i) 'faded' ii) 'abed'
iii) 'feed'
d) Assuming that originally the characters are represented using UNICODE, i.e. 16-bits, calculate the compression ratio for the strings in part (c)
e) Identify which string has the best compression ratio, and explain why.

2. Consider a Huffman tree constructed using the following binary tree class.

```
Public class BinaryTreeNode
{
  // package visible data members
  Object element;
  BinaryTreeNode leftChild;
  BinaryTreeNode rightChild;

  // … Constructors etc.

   boolean isLeaf();
}
```

Write a function `public char decompress(String s, BinaryTreeNode b)` that takes a compressed string of '0' and '1' characters, and corresponding Huffman tree `b` and returns the corresponding character, e.g. `decompress("010",someTree) = 'd'.`

3. Perform LZW compression on the following string: "TWEET_TWEET", using an initial dictionary:

0	1	2	3
E	T	W	_

Show
a) The dictionary constructed during compression
b) The compressed string

4. LZW decompression starts with an initial dictionary and works as follows

```
While there are still dictionary codes to decompress {
    let s1 = string corresponding to the next code
    let s2 = string corresponding to the following code
    output s1
    add s1 plus the 1st character of s2 to the dictionary
}
```

Using the dictionary from question 3 above, uncompress the following string:
2013468
Show the dictionary and the uncompressed string after
i) the first code has been uncompressed
ii) the second code has been uncompressed
iii) the fourth code has been uncompressed
iv) the entire string has been uncompressed.

5. a) What ISO MPEG standards exist and what are their capabilities?
b) Is MPEG a lossy compression method? If so, carefully explain what data is lost.