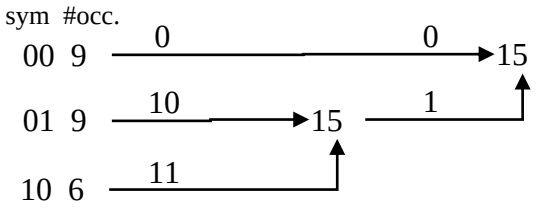
No : Answer **Name : Solution**
Eskişehir Osmangazi University, Faculty of Engineering and Architecture
Department of Electrical Engineering & Electronics, "*Communications*" Midterm 25.04.2023

Upload until : 14:59

Let a binary file is created by serializing the *bcd* codes of your 12-digit *student id*. Find the Huffman dictionary for the binary pairs (2nd ext.) in this file. Compress the file using this Huffman dictionary.

An example solution :

In this solution example, we will assume a specific *studentid*, 151220181999, since the Huffman binary-tree shapes according to the symbol probabilities. Converting the digits to *bcd* codes we get;
S_1="0001 0101 0001 0010 0010 0000 0001 1000 0001 1001 1001 1001" with 48 bits or 24 bit-pairs as S_2="00 01 01 01 00 01 00 10 00 10 00 00 00 01 10 00 00 01 10 01 10 01 10 01".
The bit-pair counts are then found as; #00=9, #01=9, #10=6, #11=0. We see that, in our example, 11 does not occur. This is normal since, for a limited length source, we cannot expect extended symbols to occur with the statistical properties of an infinite memoryless binary source. (note: in your *studentid* case, it might be a different story).
Huffman-tree is constructed according to the statistics (or occurrence counts) and dictionary is found as shown below.



Huffman dictionary

00	0
01	10
10	11

(we do not have 11 in our example)

The compressed stream is then found by replacing bit pairs in
S_2="00 01 01 01 00 01 00 10 00 10 00 00 00 01 10 00 00 01 10 01 10 01 10 01"
with their variable correspondents in the dictionary as
S_3="0 10 10 10 0 10 0 11 0 11 0 0 0 10 11 0 0 10 11 10 11 10 11 10".
The length of the compressed stream/file is seen to be 39 which is less than initial 48.
File is compressed, but the final file will have some overhead because one also needs the dictionary itself to recover the original. For such small files, the compressed file will be larger than the original because of that overhead (inclusion of the dictionary).