Solutions to Quiz 1

Question 1

Dolphins generate sounds with frequency components up to 150 kHz (or 200 kHz). What minimum sampling frequency would be required to digitize dolphin sounds?

Solution

The minimum sampling rate is twice the highest frequency, in this case $\boxed{2 \times 150 = 300 \text{ kHz}}$ (or $\boxed{2 \times 200 = 400 \text{ kHz}}$).

Question 2

A sound card manufacturer claims their hardware uses 20 (or 24)-bit samples. What would be the approximate quantization SNR when using their sound card?

Solution

The approximate quantization SNR is $6B$ dB where B is the number of bits per sample. In this case the SNR would be $\boxed{6 \times 20 = 120 \text{ dB}}$ (or $\boxed{6 \times 24 = 144 \text{ dB}}$).

Question 3

The 32-bit value **`0x12345678`** (or **`0x87654321`**) is transmitted with the bytes in big-endian order. The bits in each byte are transmitted lsb-first. What are the first 16 bits transmitted? Give your answer as a sequence of bits (0's and 1's).

Solution

The first two bytes (16 bits) transmitted in big-endian order would be **`0x12`** and **`0x34`** (or **`0x87`** and **`0x65`**). In binary, in msb-first order these are **`0001 0010`** and **`0011 0100`** (or **`1000 0111`** and **`0110 0101`**). In lsb-first order the sequence of bits transmitted would be $\boxed{\texttt{0100 1000 0010 1100}}$ (or $\boxed{\texttt{1110 0001 1010 0110}}$).

Question 4

The Runic alphabet used by the Vikings contained the letter "kaun" (ᛕ) (or "ear" (ᛠ) which has a Unicode code point of **`U+16B4`** (or **`U+16E0`**).

Find the UTF-8 encoding for this character. Give the byte values in hex. Show your work.

Solution

The value of the Unicode code point is too large for a single (values ≤ 127) or a two-byte encoding (values ≤ 2047) but a 3-byte encoding can be used (value ≤ 65535). The code point value in binary is:

`U+16B4` = **`0001 0110 1011 0100`** (or **`U+16E0`** = **`0001 0110 1110 0000`**)

Dividing the bits into groups of 4, 6 and 6 bits we get: **`0001`** (0x01), **`011010`** (0x1A), and **`110100`** (0x34) (or **`0001`** (0x01), **`011011`** (0x1B), **`100000`** (0x20)) and adding the appropriate prefixes (0xE0, 0x80, 0x80) the UTF-8 encodings are: $\boxed{\texttt{0xE1, 0x9A, 0xB4}}$ (or $\boxed{\texttt{0xE1, 0x9B, 0xA0}}$).