**1/ Describe different types of writing systems and how they vary in their use of phonetic and semantic properties. Include at least two different writing systems and limit your answer to 300 words. [3 points]**

Alphabetic system is a writing system in which each character (typically letters) represents a single sound or phoneme. In this system, characters could represent all sounds (vowels and consonants) or in some cases consonants only (also called Abjads) such as in Hebrew language. Principally it has a one-to-one sound-letter correspondence but exceptions such as multiple-one relation are also observed. English is an iconic example. In this system, characters represent sounds, emphasising the phonetic properties. And semantic meanings are not captured by or varied between letters.

Syllabic system is a writing system where each character or symbol represents a whole syllable, which usually consists of a consonant and a vowel or a single vowel. It has two main variants. One is the abugida which organises syllables systematically into families where each family shares a common consonant but varies in vowels. An example is the Sino-Tibetan language Burmese. Another one is the syllabary system where each syllable has a unique symbol but without systematic organisation. An example will be Japanese. This system focuses on the phonetic properties instead of semantic ones, more specifically, it emphasises on the syllabic sound patterns.

Logographic system is a writing system where each character or symbol represents an entire word or morpheme and carries a specific meaning. Chinese is a typical example. This system prioritises semantics over phonetics. Characters convey meaning directly, allowing for a direct representation of the semantic content.

Hybrid system is a writing system that combines elements of different systems. In this system, individual alphabetic characters could form the syllabic characters together (but not the only case). Korean is a typical example which has a blend of all three (alphabetic, syllabic, logographic) elements. This system keeps a balance and flexibility of phonetic and semantic properties.

*(Word Count: 291)*

**2/ How do different encoding systems, such as ASCII and Unicode, address the challenge of storing and representing characters in a computer’s memory, considering factors like the number of bits used and the potential for misidentification? Limit your answer to 300 words. [3 points]**

ASCII is an early system for storing English text. It addresses the challenge of sorting and representing characters by using a 7-bit binary code representation, allowing for a maximum of 128 characters. It establishes a numerical order for alphabetic sequence, simplifying character comparisons. This encoding design limits the number of bits used (at least less than Unicode) and thus keeping a balance between storage efficiency and the text representation. However, ASCII lacks support for non-English characters, thus different modifications were established in different languages, leading to potential misidentification.

Unicode, on the other hand, could cover numerous languages not only restricted to English. It is a more modern encoding system which provides a larger coding space for each character by using 32 bits of binary code for over four billion characters. However, more bits used results in higher store space consumption, especially for languages with fewer characters, such as English. To address this issue, Unicode introduced three different versions, UTF-32, UTF-16, UTF-8 with 32, 16, 8 bits per character respectively to offer more compact encoding. Among them, UTF-8 uses variable-length encoding and the highest bit as flag to minimise space usage, representing characters more efficiently while keeping flexibility between byte used and character represented. The flexible and efficient encoding of Unicode helps reduce the risk of misidentification.

*(Word Count: 224)*

**3/ Convert the following binary numbers to hexadecimal and vice versa. Show all the steps involved in the conversion. Clearly show the conversion process for each binary-to-hexadecimal and hexadecimal-to-binary conversion. [4 points]**

* **1100110010101101**

4 groups: 1100 1100 1010 1101

[1101] = 13 = D

[1010] = 10 = A

[1100] = 12 = C

[1100] = 12 = C

Thus CCAD

* **1010101101110011**

Group into 4-bit: 1010 1011 0111 0011

[0011] = 3

[0111] = 7

[1011] = 11 = B

[1010] = 10 = A

Thus A B 7 3

* **1A3F**

1 = 0001

A = 1010

3 = 0011

F = 1111

Thus 0001 1010 0011 1111

* **B7E9**

B = 11 = 1011

7 = 0111

E = 14 =1110

9 = 1001

Thus 1011 0111 1110 1001