**CS 1111-01**

**Kelechi Nwa-uwa**

**Assignment Activity Unit 2**

**Question 1**

(231)10 is to be converted to binary, octal, and hexadecimal representations. Converting a number from decimal to another base can be accomplished via successive division. This starts by dividing the decimal number by the radix of the target representation.

| **Binary (Base 2)** | **Octal (Base 8)** | **Hexadecimal (Base 16)** |
| --- | --- | --- |
| 231 ÷ 2 = 115 R 1 | 231 ÷ 8 = 28 R 7 | 231 ÷ 16 = 14 R 7 |

Then, in each iteration until a quotient of 0 has been reached, the remainder of the division is stored for later reading, while the quotient is divided by the radix.

The coefficients for binary representation are 0 & 1; for octal representation are 0, 1, 2, 3, 4, 5, 6, & 7; for decimal representation, which is used in everyday number representation, 0, 1, 2, 3, 4, 5, 6, 7, 8, & 9; and hexadecimal representation makes use of numeric and non-numeric coefficients, 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, & F.

| 115 ÷ 2 = 57 R 1  57 ÷ 2 = 28 R 1  28 ÷ 2 = 14 R 0  14 ÷ 2 = 7 R 0  7 ÷ 2 = 3 R 1  3 ÷ 2 = 1 R 1  1 ÷ 2 = 0 R 1 | 28 ÷ 8 = 3 R 4  3 ÷ 8 = 0 R 3 | 14 ÷ 16 = 0 R 14  ⇒ 0 R E |
| --- | --- | --- |

When interpreting the stored remainders, the first remainder is the least significant digit (/bit in binary representation) (LSD/B), while the last remainder is the most significant digit/bit (MSD/B). Representations are read from most significant digit/bit to the least.

| ⇒ 11100111 | ⇒ 347 | ⇒ E7 |
| --- | --- | --- |
| ∴ (231)10 = (11100111)2 | ∴ (231)10 = (347)8 | ∴ (231)10 = (E7)16 |

**Question 2**

Number encodings form the basis of the programs that interact with the computer system, such as Assembly code. An understanding of number system conversions will enable working with these programs.

Working with rotary encoders requires an understanding of Gray-Binary code conversions (Omron, n.d.). Rotary encoders are integral in robotics and in the integration of mechanical features with computer systems.

**Question 3**

Around the commencement of the computer age in the 60’s, EBCDIC and ASCII were the two main rival schemes. Currently, Unicode is becoming the standard for many purposes. I will compare these three popular coding representations (Robertson, 2020).

The American Standard Code for Information Interchange (ASCII) is the most widely used code. A 7-bit code, it is limited to the representation of 128 symbols; 34 of which are used for meta information and transmission control, and the rest for the Latin alphabet, Arabic numerals and a few other symbols (Bawden & Robinson, 2022; Robertson, 2020). By virtue of its history, it has wide adoption and is compatible with most systems.

The Extended Binary-Coded Decimal Interchange Code (EBCDIC) is an 8-bit code, primarily for IBM mainframe systems (Ndjountche, 2016; Robertson, 2020). Being an 8-bit code, it is efficient, however, adoption has hardly grown beyond IBM machines. This makes it not very compatible with most systems.

The Universal code (Unicode) in its original full-form is a 16- or 32-bit code, but with alternative encodings allowing for 8 bits. It is a much larger character set than ASCII, including characters of non-Latin alphabets, and even non-alphabetic languages. Its UTF-8 encoding, which represents the ASCII set in 8 bits, allows access to a large character set, and retains the efficiency of an 8-bit code (Robertson, 2020).

The best options for most modern systems are ASCII and Unicode, depending on whether efficiency or character set is more prioritized. For most cases, Unicode’s UTF-8 is the best bet.

**Question 4**

Omron (n.d.) defines rotary encoders as “sensors that detect position and speed by converting rotational mechanical displacements into electrical signals and processing those signals.” They are used to monitor and/or control various mechanical systems, including conveyor-belts, textile machinery, and flight simulators (Quantum Devices, 2025).

Gray code, according to Ndjountche (2016), is a non-weighted code in which only a single bit changes value between sequential transitions. It is used in rotary encoders, especially absolute encoders, because the likelihood of misreadings increases with the number of bits that change logical states between consecutive positions.

Choosing a suboptimal coding representation for the code plate of a rotary encoder, such as pure binary representation, may result in a false reading when the code changes from, say 0111 to 1000. In a practical sense, this could cause a glitch, leading to system malfunction.

**References**

Omron. (n.d.). *Technical explanation for rotary encoders: Rotary TG‑E 7.2; Rotary connect CG‑E 1.1* [PDF]. Omron.<https://www.ia.omron.com/data_pdf/guide/34/rotary_tg_e_7_2_rotary_connect_cg_e_1_1.pdf>

Quantum Devices. (2025). Rotary encoder basics: Types, uses & options. Retrieved from<https://www.quantumdev.com/resource-library/rotary-encoder-basics/>

Ndjountche, T. (2016). Digital electronics 1 : Combinational logic circuits. John Wiley & Sons, Incorporated.

Robertson, S. (2020). B c, before computers : On information technology from writing to the age of digital data. Open Book Publishers.

Bawden, D., & Robinson, L. (2022). Introduction to information science. Facet Publishing.