***Supporting Topics***

**Data Formats**

*Introduction*

Data formats are essential in the world of computing, providing a standardized way to represent and manipulate information. One of the most fundamental data formats is binary, which uses a base-2 numbering system composed of only two digits: 0 and 1. Binary is the foundation of all digital systems, with each digit representing the state of an electronic switch. Hexadecimal (hex) is another commonly used data format, employing a base-16 numbering system. It extends beyond the limited set of digits in decimal, introducing the additional symbols A-F to 0-9. Hexadecimal is frequently used to represent and display binary data in a more compact and human-readable format. On the other hand, decimal is a base-10 numbering system that most people are familiar with, utilizing ten digits 0-9. Decimal is widely used in everyday life and is the standard format for representing numerical values. Understanding these data formats is crucial for effectively working with data in various computing applications and systems.

*Task*

1. Perform a free (re-)search and explore the answers for the following questions:
   * Digits in decimal numbers are 0-9. What are the digits in hexadecimal format? What are the digits in binary format?

|  |
| --- |
| The hexadecimal format is a numbering system with base 16. There are 16 symbols ore possible digit values from 0 to 9, followed by six alphabetic characters A, B, C, D, E and F. used to represent decimal values from 10 to 15 in single bits. Binary format uses a numbering system with only two possible digits, 0 and 1. |

* + Convert (manually) the following decimal numbers to hexadecimal and binary: 8, 10, 15, 21, 32, 64, 256, 500, 512, 1000.

|  |  |  |
| --- | --- | --- |
| 8  10  15  21  32  64  256  500  512  1000 | 8  A  F  15  20  40  100  1F4  200  3E8 | 1000  1010  1111  10101  100000  1000000  100000000  111110100  1000000000  1111101000 |

* + How does Python represent these data formats? How can you use Python to convert these data formats to each other?

|  |
| --- |
| In Python, you can simply use the bin() function to convert from a decimal value to its corresponding binary value. And similarly, the int() function to convert a binary to its decimal value. The int() function takes as second argument the base of the number to be converted, which is 2 in the case of binary numbers. |

1. Use Python to:
   * Convert the decimal number 45 into its binary representation.

|  |
| --- |
| 0101101 |

* + Convert the binary number 1010101 into decimal form.

|  |
| --- |
| 85 |

* + Add the binary numbers 10111 and 1101 and express the result in binary.

|  |
| --- |
| 381 |

* + Convert the decimal number 255 into its hexadecimal representation.

|  |
| --- |
| 0xff |

* + Convert the hexadecimal number 2A into decimal form.

|  |
| --- |
| 42 |

* + Add the hexadecimal numbers C4 and 3A and express the result in hexadecimal.

|  |
| --- |
| FE |

* + Convert the binary number 1101 into decimal form.

|  |
| --- |
| 170 |

* + Convert the hexadecimal number F0 into decimal form.

|  |
| --- |
| 240 |

* + Add the decimal numbers 123 and 456.

|  |
| --- |
| 579 |

* + Convert the decimal number 157 into binary and then into hexadecimal.

|  |
| --- |
| 9D = 010011101 |

* + Convert the binary number 11101101 into decimal and then into hexadecimal.

|  |
| --- |
| 11101101 = 237 = ED |

* + Convert the hexadecimal number AB4 into decimal and then into binary.

|  |
| --- |
| AB4 = 2740 = 0101010110100 |

1. Real-life Applications:
   * Research and identify a real-world example where binary data is used extensively.

|  |
| --- |
| Binary data is used by computers in several areas; such as generating statistics, mathematical computation, and of course, computer science. It is essentially the primary language of computing systems. The data itself is identified as binary because it only uses two possible states, 0 or 1 |

* + Investigate how hexadecimal is used in computer memory addressing.

|  |
| --- |
| Hexadecimal digits are used to represent memory addresses and data in a less painful way than it is to do with binary digits.  Computers work in binary system (base 2). The main reason for choosing hexadecimal (base 16) over decimal (base 10) is how easy it is to convert from binary to hexadecimal and vice versa.  For example, a simple memory register can contain 1 byte (8 bits) of binary digits. If this register contains a value of 165 (decimal), the physical bit pattern would look like 10100101. Programmers like it to be a little bit more readable and write it as %1010\_0101.  1 byte is a group of 8 bits. Nibbles are even smaller than bytes - they are groups of 4 bits. The nice thing about hexadecimals is that 1 hexadecimal digit is the same as 1 nibble, which is four bits. The above byte pattern %1010\_0101 could be broken down and written as $A5, which is more concise.  Just as the decimal digits 165 are not written anywhere in memory, the hexadecimal digits A5 is also not stored. It's just easier to work with longer bit strings in hexadecimal. |

* + Explore how decimal data formats are used in financial calculations or accounting systems.

|  |
| --- |
| Most financial systems require decimal formatted math, to extract great precision in the values. Otherwise, there will be huge unrecognizable numbers within calculations. Decimal formats help creating readable and precise values within financial calculations. |