|  |  |  |  |
| --- | --- | --- | --- |
| Instructor |  | Due Date |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Part | **1** | **2** | **3** | **4** | Total |
| *Maximum Points* | **25** points | **25** points | **25** points | **25** points | **100**G101010 pointsG |
| ***Your Score*** |  |  |  |  |  |

**Textbook Reading Assignment**

Thoroughly read Chapter(s) 2 in your Computer Architecture and Organization textbook.

**Part 1 Glossary Terms - Data Representation**

Define, in detail, each of these glossary terms from the realm of computer architecture and organization, in general. If applicable, use examples to support your definitions. Consult your notes or course textbook(s) as references or the Internet by visiting Web sites such as:

[**http://www.bing.com**](http://www.bing.com) or [**http://www.webopedia.com**](http://www.webopedia.com/)

**(a) ASCII**

|  |
| --- |
| **The American standard code for information interchange (ASCII) is a 7-bit character encoding table that encodes common characters and machine instructions (encoded in the range 0-127). Anglo-centric, ASCII encoding doesn’t consider characters from other alphabets (e.g Mandarin) and so most systems these days are encoded in Unicode (which maintains backwards compatibility with ASCII).** |

**(b) BCD**

|  |
| --- |
| **Binary Coded Decimal (BCD) is a binary/decimal encoding common in many electronics such as alarm clocks and calculators. It simply assigns a binary number to each number in the base 10 (decimal) numeric system (i.e., 0-9). Encoded BCD numbers can be positive, negative, or unsigned, by encoding the last nibble (4 bits) in the last byte.** |

**(c) Error Detection and Correction**

|  |
| --- |
| **As transmission rates of data are increased, and more bits are packed into each square mm of storage, error rates increase. It is impossible to prevent any errors from happening at all, but several techniques (cyclic redundancy check / self-checking code, and Hamming codes, for example) exist to keep error rates in check.** |

**(d) Floating - Point Number**

|  |
| --- |
| **A floating point number is a scientific notation (i.e., 123456789 \* 10^-9) written into binary format, and contains a signed bit, a binary exponent, and a binary fractional part (or “significand”) which the binary exponent operates on. Storing numbers in this way allow for a much greater magnitude of numbers, and allow us to emulate fractional numbers with reasonable accuracy in an integer system.** |

**(e) Unicode**

|  |
| --- |
| **Unicode is a 16bit character encoding system that uses hexadecimal representations of characters from a much broader set of characters and languages than ASCII.** |

**Part 2 Exercises - Data Representation**

Provide a brief but complete answer for each of these exercises or find the solution of the given problem.

**(1)** Perform the following base conversions using subtraction or division - remainder.

(a) 588 10 = 210210 base 3

(b) 2254 10 = 33004 base 5

(c) 652 10 = 1621 base 7

(d) 3104 10 = 4228 base 9

**(2)** Represent the following decimal numbers in binary using 8 - bit signed magnitude, one's complement and two's complement and excess - 127 representation.

(a) 77

signed magnitude: 01001101

one's complement: 01001101

two's complement: 01001101

excess - 127: 11001100

(b) − 42

signed magnitude: 1101010

one's complement: 1010101

two's complement: 1010110

excess - 127: 01010101

(c) 119

signed magnitude: 01101011

one's complement: 01110111

two's complement: 01110111

excess - 127: 011110110

(d) − 107

signed magnitude: 11101011

one's complement: 10010100

two's complement: 10010101

excess - 127: 010100

**Part 3 Exercises - Data Representation**

**(1)** Suppose a computer uses 4 - bit one’s complement representation. Ignoring overflows, what value will be stored in the variable j after the following pseudocode routine terminates?

**It seems the values given in the table were incorrect? Also, the given while loop would never terminate. Going to populate the table with what I think should be the correct 4 bit binary values of j and k.**

|  |  |  |  |
| --- | --- | --- | --- |
| **j** | **Signed binary** | **k** | **Signed binary** |
| **0** | **0000** | **-3** | **1011** |
| **-1** | **1001** | **-2** | **1010** |
| **-2** | **1010** | **-1** | **1001** |
| **-3** | **1100** | **0** | **0000** |

**(2)** Assume a 24 - bit word on a computer. In these 24 bits, we wish to represent the value 295 .

1. How would the computer represent the string 295 ?  
   0000 0000 0000 0001 0010 0111

(b) If our computer uses 8 - bit ASCII and even parity, how would the computer represent the string 295 ?

50 – 57 – 53 (00110010 00111001 00110101)

(c) If our computer uses packed BCD with zero padding, how would the computer represent the number + 295 ?

Binary Value: 0000 0000 0000 0001 0010 0111

ASCII string: 50 – 57 – 53 (00110010 00111001 00110101)

Packed BCD: 0010 1001 0101 1100

**Part 4 Exercises - Data Representation**

**(1) ( Mayan Number Systems )**

The ancient Mayans used a base 20 number system. Research their particular number system and then answer each of these:

**(a)** Convert the base ten number 2010 into a Mayan numeral.

char(53)char(65)char(65)

**(b)** Convert the Roman Numeral MCXXVII into a Mayan numeral.

char(51)char(50)char(55)

**(c)** In your opinion, what were some advantages and disadvantages of the Mayan Number System.

Benefits: only a few symbols, disadvantages: converting to and from binary for computing purposes