Unit 1 – Information Representation - Activities

(1) Convert to decimal the following values:

a) $1001_2 \rightarrow 9_{10}$
b) $110010_2 \rightarrow 50_{10}$
c) $1010_2 \rightarrow 10_{10}$
d) $100101,101_2 \rightarrow 37,625_{10}$
e) $1011_2 \rightarrow 11_{10}$

(2) Convert to binary the following values:

a) $8_{10} \rightarrow 1000_2$
b) $512_{10} \rightarrow 1000000000_2$
c) $20,625_{10} \rightarrow 10100,101_2$
d) $255_{10} \rightarrow 11111111_2$
e) $3560,75_{10} \rightarrow 110111101000,11_2$

(3) Convert to hex the following values:

a) $100100101_2 \rightarrow 125_{16}$
b) $1000000000_2 \rightarrow 200_{16}$
c) $1001001_2 \rightarrow 49_{16}$
d) $11111_2 \rightarrow 1F_{16}$

(4) Convert to binary the following values:

a) $5A43_{16} \rightarrow 0101101001000011_2$
b) $BEA_{16} \rightarrow 101111101010_2$
c) $23A_{16} \rightarrow 001000111010_2$
d) $100_{16} \rightarrow 000100000000_2$
e) $F410_{16} \rightarrow 1111010000010000_2$

(5) Convert to octal the following values:

a) $100101_2 \rightarrow 45_8$
b) $11101_2 \rightarrow 35_8$
c) $110011_2 \rightarrow 63_8$
d) $100_2 \rightarrow 4_8$
e) $11010101_2 \rightarrow 325_8$

(6) Convert to binary the following values:

a) $521_8 \rightarrow 101010001_2$
b) $1234_8 \rightarrow 001010011100_2$
c) $100_8 \rightarrow 001000000_2$
d) $7543_8 \rightarrow 111101100011_2$
e) $111_8 \rightarrow 001001001_2$

(7) Convert to decimal the following values:

a) $F2A3_{16} \rightarrow 62115_{10}$
b) $4227_{16} \rightarrow 16935_{10}$
c) $4227_{8} \rightarrow 2199_{10}$
d) $AAFF_{16} \rightarrow 43775_{10}$

(8) Convert to hex the following values:

a) $16_{10} \rightarrow 10_{16}$
b) $427_{10} \rightarrow 1AB_{16}$
c) $255_{10} \rightarrow FF_{16}$
d) $534_{10} \rightarrow 216_{16}$

(9) Convert to octal the following values:

a) $16_{10} \rightarrow 20_{8}$
b) $427_{10} \rightarrow 653_{8}$
c) $255_{10} \rightarrow 377_{8}$
d) $534_{10} \rightarrow 756_{8}$

(10) Add the numbers 45 + 31 in binary code. Check the result by performing the conversion to decimal.

$45 + 31 = 86$; $86_{10} \rightarrow 1010110_{2}$; $45_{10} \rightarrow 101101_{2}$, $31_{10} \rightarrow 11111_{2}$;
$101101_{2} + 11111_{2} = 1010110_{2}$;

(11) ☠ Subtract the numbers 80 –46 in binary code. Check the result by performing the conversion to decimal.

$80 - 46 = 34_{10}$; $34_{10} \rightarrow 100010_{2}$; $80 \rightarrow 1010000_{2}$, $46 \rightarrow 101110_{2}$;
$1010000_{2} - 101110_{2} \rightarrow 100010_{2}$;

(12) Substract the numbers 109 –23 in binary code. Check the result by performing the conversion to decimal.

$109 - 23 = 86$; $86 \rightarrow 1010110$; $109 \rightarrow 1101101$, $23 \rightarrow 10111$;
$1101101 - 10111 = 1010110$;

(13) Multiply the numbers 30 * 6 in binary code. Check the result by performing the conversion to decimal.

$30 * 6 = 180$; $30 \rightarrow 11110$, $6 \rightarrow 110$;
$11110 * 110 = 10110100$;
$180 \rightarrow 10110100$;

(14) What is the negative representation of 58 in binary code? Give the result in sign and magnitude, 1's complement, 2's complement and Excess-K with K = 2^{n-1}, all for a value of 8-bit word.

1. Signed magnitude → 10111010
2. 1’s complement → 11000101
3. 2’s complement → 11000110
4. Excess-K → 01000110

(15) What is the decimal value of 10101010 if it is represented using Excess-K with K = 2^{n-1}?

214

(16) Perform the following logical operations:

a) NOT (10001001 OR 10111001) → 01000110
b) 11011011 XOR 10111001→ 01100010
c) 00000111 AND 11111111→ 00000111
d) 00000111 XOR 11111111 → 11111000

(17) ☠ How many bits should I need to represent the number 62?

You’ll need at least 7 bits to represent +62 (0111110). This is because you need a MSB in addition to other bits. Or only 6 bits if you are using only positive numbers (no MSB).

(18) ☠ With a 12 bits binary number, how many numbers can we represent?

2^{12} numbers = 4096 numbers.

(19) What is UNICODE? How many bits are used to encode a symbol?

A standard for aplphanumerical codification. Depending on the codification, it may need:

a) UTF8 → 1 to 6 bytes (8-48 bits)
b) UTF16 → 2 to 6 bytes (16-48 bits)
c) UTF 32 → 4 bytes (32 bits)

(20) Encode in decimal, octal and hex the phrase “Sistemas de representación” using the ASCII code. Note that the o is accentuated.

A) Message → “Sistemas de representación”
B)
C) DEC → 83 105 115 116 101 109 47 115 32 102 101 32 114 101 115 101 110 116 97 99 105 243 110
D) OCT → 123 151 163 164 145 155 141 163 040 146 145 040 162 145 160 162 145 163 145 156 164 141 143 151 363 156
E) HEX → 53 69 73 74 65 6D 61 73 20 66 65 20 72 65 70 72 65 73 65 6E 74 61 63 69 F3 6E

(21) ☠ What is the decimal value of C19E0000? The number is represented using 32 bitsIEEE754

19,75

(22) Perform the following conversions:

a) 34 TB → 3,4 * 10^7 MB
b) 1200 GB → 1,2 * 10^{-6} EB
c) ☠ 100 Mb → 1,25 * 10^4kB
d) ☠ 6Mb/s → 3,6288 * 10^3GB/week

(23) 🕮Divide the numbers 105/5 in binary code. Check the result by performing the conversion to decimal.

105 → 1101001, 5 → 101 ; 105 / 5 = 21, resto = 0; 21 → 10101;
1101001 / 101 = 10101 ;

(24) How long will it take (maximum) to download a 3.5GB movie if your Internet provider tells you that it provides 100 Mb/s ? And if they told you that the error rate is 5%,?

280s = 4 min 40s;
With 5% error rate → 294s = 4 min 54s; (?)

(25) In computer systems, when representing a color an RGB coding is used, so that any color is defined by the amount of Red, Green and Blue that it contains. The range of values each component can take depends on what is called*color depth*, but a general value could be 255 (or 11111111 or FF). In this way the red color would be represented by 255,0,0 or 1111111100000000000000000000(2o FF000016and an orange would be something like 255,128,0 or 1111111110000000000000000000000000000000000000000(2o FF8000(16.
With this information, you must draw (with colors obviously), the image that represents the binary code thatis indicated below. To do this, use the HTML code provided. The only thing you have to do is change the color by the code in each of the *<td>*of the code. By default is green (#00FF00).

1) 000000001111111100000000 → 00FF00
2) 000000001111111100000000 → 00FF00
3) 000000001111101000101000 → 00FA28
4) 000000001111101000101000 → 00FA28
5) 000000001111111100000000 → 00FF00
6) 000000001111111100000000 → 00FF00

7) 000000001111111100000000 → 00FF00
8) 000000001111101000101000 → 00FA28
9) 101010101010101011111111 → AAAAFF
10) 101010101010101011111111 → AAAAFF
11) 000000001111101000101000 → 00FA28
12) 000000001111111100000000 → 00FF00

13) 000000001111111100000000 → 00FF00
14) 000000001111101000101000 → 00FA28
15) 101010101010101011111111 → AAAAFF
16) 101010101010101011111111 → AAAAFF
17) 000000001111101000101000 → 00FA28
18) 000000001111111100000000 → 00FF00

19) 000000001111111100000000 → 00FF00
20) 000000001111101000101000 → 00FA28
21) 000000001111101000101000 → 00FA28
22) 000000001111101000101000 → 00FA28
23) 000000001111101000101000 → 00FA28
24) 000000001111111100000000 → 00FF00

25) 000000001111111100000000 → 00FF00
26) 111011101110111010101010 → EEEEAA
27) 111011101110111010101010 → EEEEAA
28) 111011101110111010101010 → EEEEAA
29) 111011101110111010101010 → EEEEAA
30) 000000001111111100000000 → 00FF00

31) 111011101101110101100110 → EEDD66
32) 000000001111111100000000 → 00FF00
33) 111011101101110101100110 → EEDD66
34) 111011101101110101100110 → EEDD66
35) 000000001111111100000000 → 00FF00
36) 111011101101110101100110 → EEDD66

(26) To save memory many programmers use a “word” (the minimum storage unit in memory that has a computer system) to indicate a state for each bit in the word.For example, if the word is 4 bits you could use the first one to know if the player is alive, the second one to know if he plays with a keyboard ora joystick, the third one to know if he has extra life and the fourth one to know if it's a network game. Thus, instead of using 16 bits (4 data to be stored, each 4-bit), it only uses 4.
The problem arises when you have to activate or deactivate one of these bits and, of course, get the value you save. With what you already know about binary operations, can you think of any way to do it easily?
For instance, with the word 0101:
How can I activate bit 1 to get the word 0111? and disable bit 0 to get the word 0110? and how can i get the state of the bit 3 (zero)?

Enable bit 1 → XOR mask (0010)

Disable bit 0 → XOR mask (0001)

Get state of bit 3 → AND mask (1000). After that you can only get 1000 or 0000.