# REVISIONS LOGIC - PROBLEM SOLVING

**EXERCICE 1 : Conversion**

**Q1.** 8610 🡪 base 7

Explanation : 86/7=12​​ សល់សំណល់2

12/7=1 សល់សំណល់5

SO= 152

Result :

**Q2.** 5CE16 🡪 base 10

Explanation : 5CE= 5\*16^2+C\*16^1+E\*16^0

=5\*256+12\*16+14

= 1280+192+14

= 1486

Result :

**EXERCICE 2 : Hexadecimal operation**

**Q1.** Use **2 methods** to do this operation

D45

+ 8A5

+ 1C0

Method 1 : 1101 0100 0101

1000 1010 0101

+ 0001 1100 0000 Method 2 : 1101 0100 0101

+ 1000 1010 0101

10111101010 10101 1110 1010

+ 0001 1100 0000

1011110101010

**EXERCICE 3 : Logic**

The alphabet is given below to help you:

a b c d e f g h i j k l m n o p q r s t u v w x y z

Find the missing letter in every series below:

**Q1.** c c d \_e\_ e f g g h

**Q2.** f g e h d i c \_j\_

**EXERCICE 4 : Bits, bytes and storage**

**Q1.** I have 230 colors to store. How many bits do I need? How many bytes do I need? **Justify**

230=2^7=128 F

-230= 2^8 =256 T

So= we need 8 bits for 230 color

For bytes: 1byte=8bites

So bits 8bits=1byte

**Q2.** If I have 10 bits, how many colors can I store? **Justify**

**10bits=1024color, because 2^10=1024**

**Q3.** For the RGB, we use 6 bytes. How many colors can we express? Explain the method but we don’t need the result.

**EXERCICE 5 : Encoding problem**

**Q1.**

Rules:

* 4 letters: A, E, O, U
* Each letter is repeated minimum 0 time and maximum 7 times.
* The letters are always in the alphabetic order: A then E then O then U

Examples:

AAAAEEEOOU

EEEUUUUUUU

AAEEOOUU

1. Find an encoding of maximum **12 bits**. Explain the method, explain the size and give examples.
2. Is your encoding lossless or loosely?

**Q2.**

Now, the letters can be in any order

Example:

EEEOOAAAAU

UUEEEOAA

1. Does the encoding you found before work for this new rule?
2. Find a new lossless encoding. What is the minimum size of this encoding? Explain and give example**s**.
3. For this example **EEEEEE**, what is the most little size possible with your encoding?
4. Is this encoding a compression ?