TPOP PRACTICAL

MODULE RECAP

Reading Week

Question 1: Strings and patterns

```
>>> printBasePattern(5)
x...x
.x.x.
..x..
.x.x.
x...x
>>>
```

Write a function `printBasePattern(n)` taking one integer parameter representing the size of the pattern to print. The size of the print must be greater or equal to 3. If the size is invalid you should print an error message (or even better raise an exception). The pattern should represent a cross as shown on the left.

Question 2: Strings and patterns

Same question as above, except that two additional string parameters must be given, each one representing the different character of the pattern.

```
>>> printExtendedBasePattern(6,'O','-')
O----O
-O--O-
--OO--
--OO--
-O--O-
O----O
>>>
```

Question 3: Strings and patterns

```
O--O
-OO-
-OO-
O--O
```

This time we want to reproduce several times the same pattern (shown on the left). Write a function `printMultipleBasePattern` taking two integer parameters, the first one representing the number of rows and the second parameter representing the number of columns. The pattern should be printed in each cells, e.g. a total of `rows` × `columns` times. Both parameters must be greater or equal to 0. If they are invalid you should print an error message (or even better raise an exception). An example of the result is given below.

```
>>> printMultipleBasePattern(2,3)
O--OO--OO--O
-OO--OO--OO-
-OO--OO--OO-
O--OO--OO--O
O--OO--OO--O
-OO--OO--OO-
-OO--OO--OO-
O--OO--OO--O
>>>
```

Question 4: Encoding and decoding messages.

Consider the following dictionaries:

```
decypherbook = {'0000':8, '0001':1, '0010':0, '0011':9,
                '0100':5, '0101':3, '0110':7, '0111':2,
                '1110':4, '1111':6}
cypherbook   = {8:'0000', 1:'0001', 0:'0010', 9:'0011',
                5:'0100', 3:'0101', 7:'0110', 2:'0111',
                4:'1110', 6:'1111'}
```

i. We can use the `cypherbook` dictionary to encode numbers such as 1253495. Write a function `encode( cyper, number)` taking a cypher dictionary and a string representing a number as parameters and returned the encoded number.

   For example `encode(cypherbook, "12")` should return "00010111".

   First assume that input parameters are correct, and then change your code to test when inputs may be incorrect.

ii. Same exercise but this time to decode a message. In this case we must pass in the parameters the dictionary to be used to decipher the message.
    For example, `decode(decypherbook, "00010111")` should return "12"

iii. For part i and ii we need to use two dictionaries, one for encoding and one for decoding. Write a function such that the same dictionary is used for coding and decoding. For example, `advanced_decode(cypherbook, "00010111")` should return "12" and `advanced_encoode(cypherbook, "12")` should return `"00010111"`.