1. I decided to implement the encryption and decryption in Python!
I'm thinking of two functions:

- `encrypt(message,shift)`
- `decrypt(encryptedMessage,shift)`

Download the **`EncryptingTheMessage.py`** file from our web page. Run it.
Try to encrypt the message "hello" with a shift of 3. You should be getting "khoor".
Try to encrypt the message "Hello, how are you?" with the same shift of 3. You should be getting "khoor6krz3duh1brx" as the result.

Your responsibility will be to work on the decryption function.
As you can see from the **`EncryptingTheMessage.py`** file a silly "undoing" the shift doesn't work.
Use the draft on the next side of this page and follow the comments:

```
def decrypt(encryptedMessage, shift):
  """ this function will decrypt the message,
  encryptedMessage is the encrypted message,
  shift is the shift in alphabet that was used to encrypt it,
  must be an integer;
  the function returns the original (decrypted) message """

  # First, make sure that the type of shift is integer
  assert type(shift) == type(1)

  # Second, we will start with an empty string
  result = ""

  # put the for loop here, iterating over all symbols in the
  # encryptedMessage:

  # the following should be in the BODY of the for loop:
    # replace the <condition1> in the if below:
    # say that ord(ch) should be between 49 and 57
    # (we encrypted the space with these ASCII codes that stand
    # for numbers 1 through 9
    if <condition1>:
      result += ' '

    # replace the <condition2> in the elif below:
    # say that ch must be greater than or equal to the letter 'a',
    # and it must be less than or equal to the letter 'z'
    # (this means that ch is a letter from the alphabet)
    elif <condition2>:

      ch_code = ord(ch)

      # when a backwards shift (to the left) hops over letter a:
      if ch_code - shift < 97:
        result += chr(123-(97-(ord(ch) - shift)))

      # when the backwards shift doesn't go more to the left than a:
      else:
        result += chr(ord(ch) - shift)

    # in case if the symbol is neither space nor a letter from the
    # alphabet, simply ignore it
    else:
      pass

# we are finished with the BODY of the for loop;
# put the return statement right after the for loop;
# let's return the resulting string.
#  Replace the question marks with what needs to be returned
  return ???


# the following code is OUTSIDE the function decrypt definition!
# it is testing how oyr decrypt function works!

# decrypting the encrypted message, the original message was 'hello'
print(decrypt("khoor",3))

# decrypting; the original message was 'Hello, how are you?'
print(decrypt("khoor6krz3duh1brx",3))
```