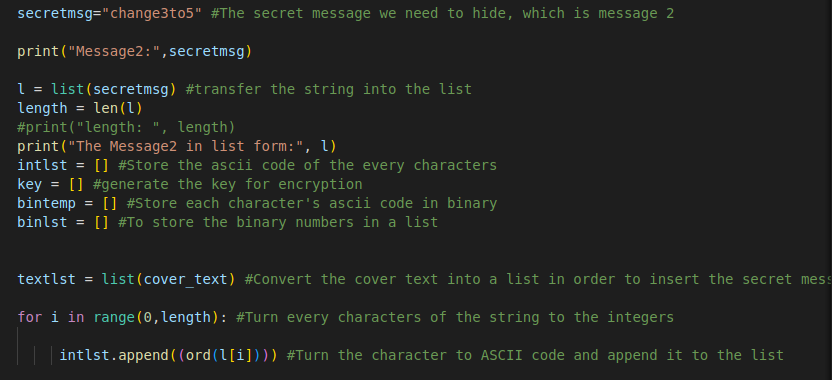
This program aims to create an application as a tool for text communications when people need to share some secret information. There are mainly two functions has been provided in The App which are Encryption and Decryption.

**Encryption**

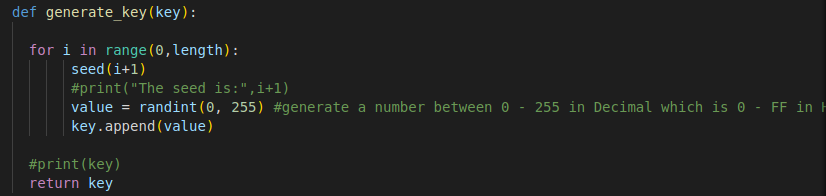
**1. Transfer each character of Message 2 to ASCII code**



**Figure 6 – Show the code that transfer the string to an ascii code list**

To encrypt the message, it is necessary to convert each character in Message 2 to ascii code, I transfer the message2 string into a list and used a for loop to turn every character the ascii code, and these ascii code will be stored in a list. **(Figure 6)**

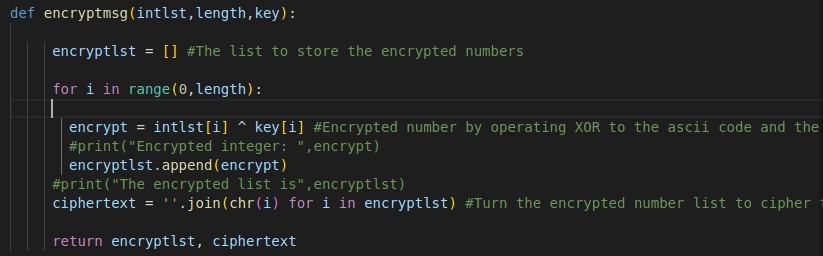
**2. Generate the key**



**Figure 7 – Generate a key that for the encryption**

The key should be generated in order to operate the encryption process by XOR cipher. the key will be generated by pseudo random generator and which the generator will generate a number between 0 to 255 which is the range of ASCII code.

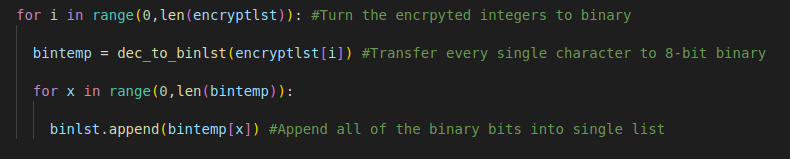
**3. XOR cipher to encrypt Message2**



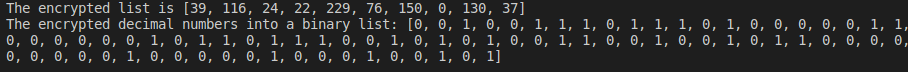
**Figure 8 – Encrypt the message with the key**

After the key has been generated, the XOR cipher process will start, and the cipher text will be generated for steganography in following steps.

**4. Turn encrypted list of Ascii to a list of 8-bits binary code**



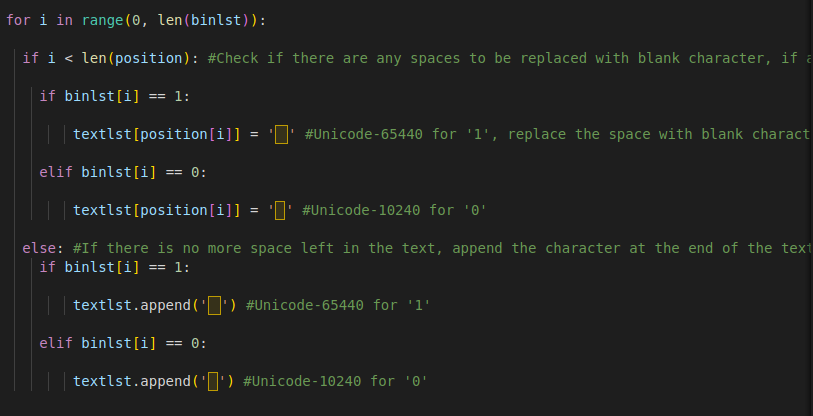
**Figure 9 – turn the list with encrypted code to 8-bits binary code**



**Figure 10 – Encrypted numbers to a list of binary numbers**

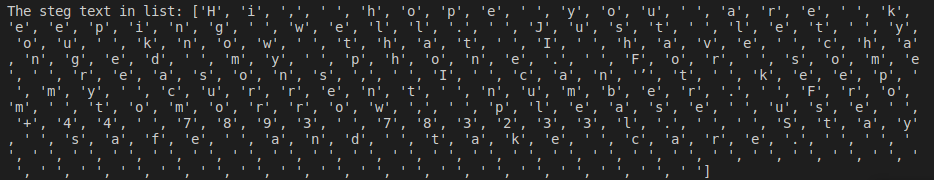
Before hide the binary bits to Message1, it is necessary to transfer all decimal ASCII codes to binary numbers, and the binary number must be in 8-bits for the decryption purpose.

**5. Hide the list of binary numbers to Message 1**



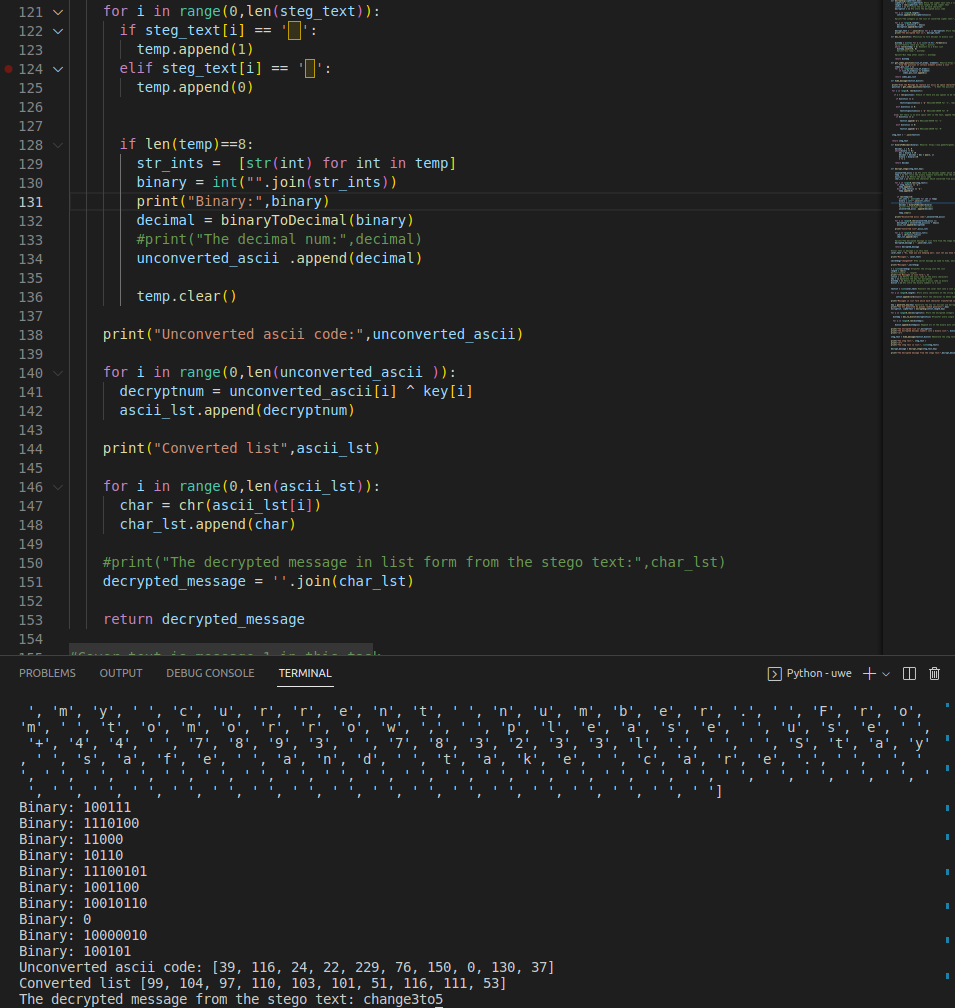
**Figure 11 – Use blank character to replace the ASCII 32 character**

The way I chose to hide the data is using whitespace character to hide the Message2 to the Message1. Whitespace character is invisible to human and there are different whitespace characters exist which human could not tell the difference between them in plain-sight, but these invisible characters had their own code, and the computer could tell the difference between them, so I used the 2 different whitespace characters Unicode-65440 and Unicode-10240 to represent ‘1’ and ‘0’. In Message1 the spaces are whitespace characters are ASCII 32, I used a function **(Figure 11)** to replace all ASCII 32 whitespace characters with Unicode-65440 and Unicode-10240 in order to hide my binary numbers into it. Also, if the message is too short there doesn’t got enough space to hide the binary numbers, I will append the bits at the end of the message to deal with the situation of message 1 is too short.



**Figure 12- The spaces have been replaced and some whitespace characters inserted at the end of the message.**

**Decryption**



**Figure 13 – Decrypt the message from the steg text**

To decrypt the steganography text, the program will regroup the binary numbers hidden in the steg text, every 8-bit of binary numbers will be grouped and convert back to decimal numbers in order to transfer it back to ASCII code and finally it will perform the XOR cipher process again and the original text will be shown.