Many Time Pad

**Done by:** Ridinbal MD, Hridai Gianchandani, Malak Djeradi, Fatma Alhai

Table of Contents

[Introduction 4](#_Toc137734073)

[Steps 4](#_Toc137734074)

[1) XOR Each ciphertext with the 13th Ciphertext 4](#_Toc137734075)

[**Python Code Snippets with Explanation:** 4](#_Toc137734076)

[2) Find the Binary Value of Each Result 6](#_Toc137734077)

[**Python Code Snippets with Explanation:** 6](#_Toc137734078)

[**Output:** 7](#_Toc137734079)

[3) Convert 8 Bits of Binary at A Time for Each Binary Value to ASCII Values 12](#_Toc137734080)

[**Python Code Snippet with Explanation:** 12](#_Toc137734081)

[**Output:** 14](#_Toc137734082)

[4) Find Key Values 16](#_Toc137734083)

[**Python Code Snippet with Explanation** 16](#_Toc137734084)

[**Output:** 18](#_Toc137734085)

[5) Printing The Complete Secret Message 20](#_Toc137734086)

[**Python Code Snippet with Explanation** 20](#_Toc137734087)

[**Output:** 20](#_Toc137734088)

[Full Python Code: 21](#_Toc137734089)

[Complete Output 26](#_Toc137734090)

# **Introduction**

The assignment is about Many Time Pad. In Many Time Pad, one key is used to encrypt multiple messages which makes it vulnerable and easy for hackers to crack it. In this assignment, we are given 13 hex-encoded ciphertexts and we need to decrypt the 13th ciphertext. To decrypt it, we need to find a key. So, in this assignment, we will be mentioning steps to find the key using a python program.

# Steps

## **1) XOR Each ciphertext with the 13th Ciphertext**

XOR each message with message 13 and you will get the result in a hex string. Also, we need to make sure all messages are of equal length to get the correct XOR result. So, trimming the last hex strings in messages which are longer and match with message 13.

### **Python Code Snippets with Explanation:**

Function to perform the XOR operation

# Function to find XOR values, it returns the result in Hex String  
def xor\_func(m1, m13):  
 xor\_hex\_value = hex(m1 ^ m13)  
 return xor\_hex\_value

We store each hex string in their respective variables named from m1 to m12.

# We trimmed Messages 1 to 12 to match the length of Message 13  
m1 = 0x0f351c71e76f5fbe548d4c54a69a2bcb3d2e4ceb3cfb5250c24dff419949683f8ed3a5f04f57116fe797410d2c138b60db6da534a7abe0f658b65b5c0ccbeb67d1c9d9216d8befeb35173f3596f40d4fa84e1e702818fbc06bec90d4315fceacfa7112c3e5 #5d74aaf3394bb08f7504a8e5019c4e3e838e0f364946f31721a49ad2d24ff6775efcb4f79fe4217a01b43cb5068bf3b52ca76543187274  
m2 = 0x1a311571ff660da658c80545e98325ca646746a22ef55202d04cf90d93067c70a6c8b6b94c161120af95421b3a138b609d6abe2ae6b2e6eb42f7431405c4a56ad1c9cf3b67cafbe72301393583e80d58a34517767b19b58166a0c3db304acfbafa721591ea #4d398af07c49b69a7118fcff4889597aca81433b4953f50b2bbbdf9dcd0aff7013d3b5a3d3ec2b73019c3aa91b8bddf411a72b480c636cc6597494151386  
m3 = 0x0f381a39fe6f41bd57c44646eacc3ecb2b695ae729ee174ac650ab0c92547a73b19ca7a24d40162bea9c0d1c2c1395678f6dec2ae8b4eaa449b1511507c3e06dd6c9c02c69c5eca4241d3f3585f44440ad011078381bacc075e4c3db304acfbafa721591ea  
m4 = 0x0f351c71d96e59b742c34053a6853d9930664da237f24456874ae61198546b7ea6c8f7a34b581823ead8490c29568e618b68a929f3e6e6ea0ca35f1944c2ec70d686df3028c4f9a42a0720748cbb4e41a74c07773213bad56eefde922d44cdbcbf3e008be8 #0870a5eb7c55ab907f18fcf347dd433bcf83432d0849e80c22b0  
m5 = 0x127d183cb07342a042d40553e9cc3dd83d2e5cea3be91756cf46f904d74f6c3fbcd3b8f04f431c27af8842003147c27a9425b82fe2e6f8ed5fb2540e05c9ee23d681cc3d28c7f6e227522466c2fe555aa34f116d7b1fb58168f4d8d72c0dd3b3bb701197fe #087baefe784eac9c3002b4f9488f0029c08606341d49ef113ff7cdd8c60abe6f5cefbff792e9317e4f9d36b948dfddf409e264580f65  
m6 = 0x1835183ce0614abc558d4c41a69521cc646f5ae77aee5247cc4ae506d752777ae8c8a5a5565e5f26fcd84f0c2b47877cdb71a426e9e6eea440be525c00cff166c19dc23b28e2eba4271c2e7a97e94c49af5252787b1dbacf27f4df923c4883baa26e158dfe #416faebd7c4dba973004b9ff4a914529d0cf1432004cf94520bedf9dd00aea6750e9b5a580ad266d44de3cb304d295f447a1634c117a68c41e67d50d09c35928a62e6d6648371b40ac83b274cecb04d6c41b6fba  
m7 = 0x1928103df46943b510d94044ee8227da256208f123ee4347ca50ab0899507073bed9a4f043161320fbd8420f7f5b837c9f25bb28f5adafe542b3170f14cfe66ac385c4336dcfbfef2c1d3a7987ff4a4bea4d13773c05bac662f390d3304983afa871008cee #4775b8bd7a54bb907e11fcfd4f99003ec68d163d0e49f2026ca3dfcec006f06513fcb4b3d3ff2279409d27b21ac2dbf2  
m8 = 0x12290a71f96d5dbd43de4c45ea896ecd2b2e45ed2cf81756c803e70881433f6ba79cb8a047441e3bead84c1d7f528c77db69a931e2aaaff345a35f1311dea56fc788db2066ccbff030132e7091bb4f47be52526a3e15b6c869e7dccb7e40c6beb4771a84e1 #4d6ab8bd7f49be9e7d13b2e852dd4f3c839f06281a4ff20420f7d3d3d200ec6f52e9b3b89d  
m9 = 0x0f351c71c7654ff251de056ea68920cf2d7d49e53ff9174bd303fc04d74e7e69ad9cb9bf56160c2aea960d002b139b6b8f25982fe2e6e9f158a2451944c3f623d19dc425648beceb621f38768abb4f47ad46176b7b04b3c069a0c4da3b0dd3bea96a54b7e4 #453989f86b55ba8b635b90f944  
m10 = 0x127d0c22f5640da65f8d514fef822599306649f67afe4e40c251f81196457a3fbfdda4f0445f193bf6d8540c3e41912e9a72ad3ea791e7e558f77e5c10c2ea76c581d9697fcaeca4241b2b619bbb544bab5301393a07bad827f7d1c17e42cdb3a33e0086e3 #0860aefc6b48ff986717a5bc6093447ad487022e4969bc1124b8cfdadc1bbe7552eefaa396e36766449f21ae48cac2f41ee262595d616cd95963990b03824934ea2a287844722140b583a2638bcf16c3df0323a212740f3d5b517fbd10e4e253512e  
m11 = 0x0c385930e2650da658c80544ee8522dd366b46a235fb17438757ee029f487073a7dbbeb3435a5f2ee89d0d3e3a138a6f8d60ec21e8b3e1e00ca4430e01cbe86fcb87c82d28dcfefd31522273c2ff4247a44652742e13b38168e690dd2b5f83adb56b008ae3 #4d39bcf26b50ffa9621fb2e84893477aca9c43340600f00a22b0dfcf941bf66713f2b4bb8aad307e58de3cbb48d9d0e515ad6f581e7f63cd59609a160d900d1faf2329634f354814b793bc37c3d700d5c71271e30d740e7815516dbd1af8a344533fcb  
m12 = 0x0829003df52058a155c90553e9cc2cdc646f46a233f34347d542e8159e49713faad9a3a74753116ffb90484937468f6f9525bf28f2aaafe542b317080bc5e970829dc5287c8be8e130176d798bf6445aa34f1539121efbd56fe590d6374acaabbb725486ff #4939a2e9394cb6957c56b4fd5798002ecccf00350445bc033eb8d79dc007fb2240f2afbbd3ec2b704f9b  
  
m13 = 0x14331c71fd614eba59c34007e58d2099206108f632f81755c851e04198403f79a1daa3a902590d2be6964c1b26138f6b95258228a7abeee744be591944c9e46d828dc2697cc3faa4351d3f7ec2f44b0ea54f17393e08afd366efc2d63743c2ada33e1982e3

Then we store all of it in list\_holder1 which is a list and now it contains different hex strings.

# Storing All Messages  
list\_holder1 = [m1, m2, m3, m4, m5, m6, m7, m8, m9, m10, m11, m12]

We call the function inside a for loop to perform the XOR operation (each message in list\_holder1 with message 13) and store the hex string result in a variable called hex\_xor\_value.

for i in range(0, 12):  
 # Calling the XOR Function  
 xor\_hex\_value = xor\_func(list\_holder1[i], m13) ##List\_holder1 is a list that contains all messages and each messages xored with message 13 using hex\_function and the value is stored in xor\_hex\_value  
 print(“Hex value of Message”, i + 1, “XOR’d with Message 13:”)  
 print(xor\_hex\_value + “\n”)

## **2) Find the Binary Value of Each Result**

Once we get the result in a hex string when we XOR each message with the 13th message. We find the binary value of each result.

### **Python Code Snippets with Explanation:**

Function to convert each result to binary

# Function to convert the above-found Hex String to Binary  
def hex\_to\_binary\_func(xor\_hex\_value):  
 hex1 = xor\_hex\_value[2:]  
 binary\_value = bin(int(‘1’ + hex1, 16))[3:]  
 return binary\_value

This is list\_holder3 which is an empty list, and it will later contain binary values.

list\_holder3=[]

This code is inside the same loop mentioned before in the previous step. hex\_to\_binary\_func is a function that is called to convert hex value to binary and the value is stored in binary\_value. There is an if condition implemented to add an extra four 0’s to the binary values in index 1, 4, 5, 6, 7, and 9 to get the correct binary value. Binary values are appended to a list called list\_holder3.

# Calling the Hex String to Binary Function  
binary\_value = hex\_to\_binary\_func(xor\_hex\_value)  
print(“Binary of Message”, i + 1, “XOR’d with message 13:”)  
  
# Adding four 0’s to the Binary of XOR’d Hex’s of messages 1, 4, 5, 6, 7, 9  
# because all these messages start with 1 instead of 0  
if(i == 1 or i == 4 or i == 5 or i == 6 or i == 7 or i == 9):  
 binary\_value = “0000” + binary\_value  
 print(binary\_value + “\n”)  
 list\_holder3.append(binary\_value) #Storing binary values to list\_holder3  
  
else:  
 print(binary\_value + “\n”)  
 list\_holder3.append(binary\_value) #Storing binary values to list\_holder3  
  
print(“---------------------------------------------------------------------------------------------------------------” + “\n”)

### **Output:**

Hex value of Message 1 XOR’d with Message 13:

0x1b0600001a0e11040d4e0c5343170b521d4f441d0e0345050a1c1f00010957462f0906594d0e1c4401010d160a00040b4e48271c00000e111c08024548020f0a53441b481148154f000a004b540046410d010949161054130d035202061c0c01594f0b4106

Binary of Message 1 XOR’d with message 13:

0001101100000110000000000000000000011010000011100001000100000100000011010100111000001100010100110100001100010111000010110101001000011101010011110100010000011101000011100000001101000101000001010000101000011100000111110000000000000001000010010101011101000110001011110000100100000110010110010100110100001110000111000100010000000001000000010000110100010110000010100000000000000100000010110100111001001000001001110001110000000000000000000000111000010001000111000000100000000010010001010100100000000010000011110000101001010011010001000001101101001000000100010100100000010101010011110000000000001010000000000100101101010100000000000100011001000001000011010000000100001001010010010001011000010000010101000001001100001101000000110101001000000010000001100001110000001100000000010101100101001111000010110100000100000110

---------------------------------------------------------------------------------------------------------------

Hex value of Message 2 XOR’d with Message 13:

0xe0209000207431c010b45420c0e055344064e541c0d4557181d194c0b464309071215104e4f1c0b49030e001c00040b084f3c024119080c06491a0d410d410753440d521b090143161c064b411c4656060a004f45111a52004f010d07090d17594c0c1309

Binary of Message 2 XOR’d with message 13:

0000111000000010000010010000000000000010000001110100001100011100000000010000101101000101010000100000110000001110000001010101001101000100000001100100111001010100000111000000110101000101010101110001100000011101000110010100110000001011010001100100001100001001000001110001001000010101000100000100111001001111000111000000101101001001000000110000111000000000000111000000000000000100000010110000100001001111001111000000001001000001000110010000100000001100000001100100100100011010000011010100000100001101010000010000011101010011010001000000110101010010000110110000100100000001010000110001011000011100000001100100101101000001000111000100011001010110000001100000101000000000010011110100010100010001000110100101001000000000010011110000000100001101000001110000100100001101000101110101100101001100000011000001001100001001

---------------------------------------------------------------------------------------------------------------

Hex value of Message 3 XOR’d with Message 13:

0x1b0b0648030e0f070e0706410f411e520b0852111b16001f0e014b4d0a14450a1046040b4f191b000c0a41070a001a0c1a486e024f1f04430d0f080c430a040054440245150616001100004b47000f4e084e074106130313130b010d07090d17594c0c1309

Binary of Message 3 XOR’d with message 13:

0001101100001011000001100100100000000011000011100000111100000111000011100000011100000110010000010000111101000001000111100101001000001011000010000101001000010001000110110001011000000000000111110000111000000001010010110100110100001010000101000100010100001010000100000100011000000100000010110100111100011001000110110000000000001100000010100100000100000111000010100000000000011010000011000001101001001000011011100000001001001111000111110000010001000011000011010000111100001000000011000100001100001010000001000000000001010100010001000000001001000101000101010000011000010110000000000001000100000000000000000100101101000111000000000000111101001110000010000100111000000111010000010000011000010011000000110001001100010011000010110000000100001101000001110000100100001101000101110101100101001100000011000001001100001001

---------------------------------------------------------------------------------------------------------------

Hex value of Message 4 XOR’d with Message 13:

0x1b060000240f170d1b00005443081d0010074554050a53034f1b0650001454070712540a490115080c4e05170f45010a1e4d2b01544d080d481d0600000b081d540b1d59540703001f1a1f0a4e4f054f0203104e0c1b150608001c441a070f111c0019090b

Binary of Message 4 XOR’d with message 13:

0001101100000110000000000000000000100100000011110001011100001101000110110000000000000000010101000100001100001000000111010000000000010000000001110100010101010100000001010000101001010011000000110100111100011011000001100101000000000000000101000101010000000111000001110001001001010100000010100100100100000001000101010000100000001100010011100000010100010111000011110100010100000001000010100001111001001101001010110000000101010100010011010000100000001101010010000001110100000110000000000000000000001011000010000001110101010100000010110001110101011001010101000000011100000011000000000001111100011010000111110000101001001110010011110000010101001111000000100000001100010000010011100000110000011011000101010000011000001000000000000001110001000100000110100000011100001111000100010001110000000000000110010000100100001011

---------------------------------------------------------------------------------------------------------------

Hex value of Message 5 XOR’d with Message 13:

0x64e044d4d120c1a1b1745540c411d411d4f541c09110003071719454f0f53461d091b594d1a110c491e0e1b17544d1101003a07454d160a1b0c0d1741000a4e540c0e5454040c46124f1b18000a1e540600065445171a520e1b1a011b4e111e184e08151d

Binary of Message 5 XOR’d with message 13:

0000011001001110000001000100110101001101000100100000110000011010000110110001011101000101010101000000110001000001000111010100000100011101010011110101010000011100000010010001000100000000000000110000011100010111000110010100010101001111000011110101001101000110000111010000100100011011010110010100110100011010000100010000110001001001000111100000111000011011000101110101010001001101000100010000000100000000001110100000011101000101010011010001011000001010000110110000110000001101000101110100000100000000000010100100111001010100000011000000111001010100010101000000010000001100010001100001001001001111000110110001100000000000000010100001111001010100000001100000000000000110010101000100010100010111000110100101001000001110000110110001101000000001000110110100111000010001000111100001100001001110000010000001010100011101

---------------------------------------------------------------------------------------------------------------

Hex value of Message 6 XOR’d with Message 13:

0xc06044d1d0004060c4e0c4643180155440e521148164512041b05474f1248034912060c5407520d1a4e03170d5408174e54260e4e4d004304000b454406150b431000525421110012011104551d07470a1d45414515151c411b1d440b0b411701500c0f1d

Binary of Message 6 XOR’d with message 13:

0000110000000110000001000100110100011101000000000000010000000110000011000100111000001100010001100100001100011000000000010101010101000100000011100101001000010001010010000001011001000101000100100000010000011011000001010100011101001111000100100100100000000011010010010001001000000110000011000101010000000111010100100000110100011010010011100000001100010111000011010101010000001000000101110100111001010100001001100000111001001110010011010000000001000011000001000000000000001011010001010100010000000110000101010000101101000011000100000000000001010010010101000010000100010001000000000001001000000001000100010000010001010101000111010000011101000111000010100001110101000101010000010100010100010101000101010001110001000001000110110001110101000100000010110000101101000001000101110000000101010000000011000000111100011101

---------------------------------------------------------------------------------------------------------------

Hex value of Message 7 XOR’d with Message 13:

0xd1b0c4c09080d0f491a00430b0f0743050300071116541202014b4901104f0a1f030759414f1e0b1d4e0e1459480c170a00390052064102060d4e16500602074108065a110c454b19000507450b01454f02044e020d1515041c5205070a41020b4f190e0d

Binary of Message 7 XOR’d with message 13:

0000110100011011000011000100110000001001000010000000110100001111010010010001101000000000010000110000101100001111000001110100001100000101000000110000000000000111000100010001011001010100000100100000001000000001010010110100100100000001000100000100111100001010000111110000001100000111010110010100000101001111000111100000101100011101010011100000111000010100010110010100100000001100000101110000101000000000001110010000000001010010000001100100000100000010000001100000110101001110000101100101000000000110000000100000011101000001000010000000011001011010000100010000110001000101010010110001100100000000000001010000011101000101000010110000000101000101010011110000001000000100010011100000001000001101000101010001010100000100000111000101001000000101000001110000101001000001000000100000101101001111000110010000111000001101

---------------------------------------------------------------------------------------------------------------

Hex value of Message 8 XOR’d with Message 13:

0x61a1600040c13071a1d0c420f044e540b4f4d1b1e000003005207491903001206461b09451d13100c4e00065941031c4e4c2b1945014114011d060a55174102450519491a0f4554050e110e534f04491b1d4553001d191b0f081e1d490304131749030602

Binary of Message 8 XOR’d with message 13:

0000011000011010000101100000000000000100000011000001001100000111000110100001110100001100010000100000111100000100010011100101010000001011010011110100110100011011000111100000000000000000000000110000000001010010000001110100100100011001000000110000000000010010000001100100011000011011000010010100010100011101000100110001000000001100010011100000000000000110010110010100000100000011000111000100111001001100001010110001100101000101000000010100000100010100000000010001110100000110000010100101010100010111010000010000001001000101000001010001100101001001000110100000111101000101010101000000010100001110000100010000111001010011010011110000010001001001000110110001110101000101010100110000000000011101000110010001101100001111000010000001111000011101010010010000001100000100000100110001011101001001000000110000011000000010

---------------------------------------------------------------------------------------------------------------

Hex value of Message 9 XOR’d with Message 13:

0x1b0600003a040148081d4569430400560d1c41130d01001e1b521c454f0e41100c461a16544f01010c00411b0d0014001a001a07454d07161c1c1c00000a124e5310064c1848164f57020708484f044908090052450c1c130f4f060c0c4e11130a544d3507

Binary of Message 9 XOR’d with message 13:

0001101100000110000000000000000000111010000001000000000101001000000010000001110101000101011010010100001100000100000000000101011000001101000111000100000100010011000011010000000100000000000111100001101101010010000111000100010101001111000011100100000100010000000011000100011000011010000101100101010001001111000000010000000100001100000000000100000100011011000011010000000000010100000000000001101000000000000110100000011101000101010011010000011100010110000111000001110000011100000000000000000000001010000100100100111001010011000100000000011001001100000110000100100000010110010011110101011100000010000001110000100001001000010011110000010001001001000010000000100100000000010100100100010100001100000111000001001100001111010011110000011000001100000011000100111000010001000100110000101001010100010011010011010100000111

---------------------------------------------------------------------------------------------------------------

Hex value of Message 10 XOR’d with Message 13:

0x64e10530805431c064e11480a0f050010074100480659150a0018500e0545461e07075946061410104e181718521e450f572f16003a09021c492745540b0e1b470c1b00030916001106141f594f1f450e1c1600040f150b4118131749010f1e0000190400

Binary of Message 10 XOR’d with message 13:

0000011001001110000100000101001100001000000001010100001100011100000001100100111000010001010010000000101000001111000001010000000000010000000001110100000100000000010010000000011001011001000101010000101000000000000110000101000000001110000001010100010101000110000111100000011100000111010110010100011000000110000101000001000000010000010011100001100000010111000110000101001000011110010001010000111101010111001011110001011000000000001110100000100100000010000111000100100100100111010001010101010000001011000011100001101101000111000011000001101100000000000000110000100100010110000000000001000100000110000101000001111101011001010011110001111101000101000011100001110000010110000000000000010000001111000101010000101101000001000110000001001100010111010010010000000100001111000111100000000000000000000110010000010000000000

---------------------------------------------------------------------------------------------------------------

Hex value of Message 11 XOR’d with Message 13:

0x180b45411f04431c010b45430b080244160a4e54070300164f060e4307084f0a06011d1a410352050e0b41251c00050418456e094f180f07481a1a1745020c02490a0a44541f0459044f1d0d000b09490109454d101b1c520e09520b1c1c41001655190800

Binary of Message 11 XOR’d with message 13:

0001100000001011010001010100000100011111000001000100001100011100000000010000101101000101010000110000101100001000000000100100010000010110000010100100111001010100000001110000001100000000000101100100111100000110000011100100001100000111000010000100111100001010000001100000000100011101000110100100000100000011010100100000010100001110000010110100000100100101000111000000000000000101000001000001100001000101011011100000100101001111000110000000111100000111010010000001101000011010000101110100010100000010000011000000001001001001000010100000101001000100010101000001111100000100010110010000010001001111000111010000110100000000000010110000100101001001000000010000100101000101010011010001000000011011000111000101001000001110000010010101001000001011000111000001110001000001000000000001011001010101000110010000100000000000

---------------------------------------------------------------------------------------------------------------

Hex value of Message 12 XOR’d with Message 13:

0x1c1a1c4c0841161b0c0a45540c410c45440e4e54010b54121d13085406094e460b03000e450a1c441d0604521155000400003d0055014102060d4e114f0c0d1d0010074100481245050a520749020f54060002002c165406090a520000090806184c4d041c

Binary of Message 12 XOR’d with message 13:

0001110000011010000111000100110000001000010000010001011000011011000011000000101001000101010101000000110001000001000011000100010101000100000011100100111001010100000000010000101101010100000100100001110100010011000010000101010000000110000010010100111001000110000010110000001100000000000011100100010100001010000111000100010000011101000001100000010001010010000100010101010100000000000001000000000000000000001111010000000001010101000000010100000100000010000001100000110101001110000100010100111100001100000011010001110100000000000100000000011101000001000000000100100000010010010001010000010100001010010100100000011101001001000000100000111101010100000001100000000000000010000000000010110000010110010101000000011000001001000010100101001000000000000000000000100100001000000001100001100001001100010011010000010000011100

---------------------------------------------------------------------------------------------------------------

## **3) Convert 8 Bits of Binary at A Time for Each Binary Value to ASCII Values**

Once we get the binary values, convert 8 bits of binary to ASCII and store it in a list.

### **Python Code Snippet with Explanation:**

Function to convert every 8 bits of binary to ASCII

# Function to convert the above Binary to ASCII  
def toString(binaryString):  
 return ““.join([chr(int(binaryString[i:i + 8], 2)) for i in range(0, len(binaryString), 8)])  
  
# Function to convert 8 bits at a time to ASCII and store it  
def ascii\_convertor\_func(binary\_value):  
 mystring = ““  
 store\_value = ““  
 list1 = []  
 i = 0  
 # print(“Printing 8 bits of Binary:”)  
  
 for x in binary\_value:  
  
 mystring = mystring + x  
  
 # Converting 8 bits to ASCII  
 if len(mystring) >= 8:  
 store\_value = toString(mystring)  
  
 list1.append(store\_value)  
  
 i = i + 1  
  
 mystring = ““  
  
 return list1

We are storing all 12 lists in list\_holder2 which will be later used in the program to store ASCII values.

list\_m1 = []  
list\_m2 = []  
list\_m3 = []  
list\_m4 = []  
list\_m5 = []  
list\_m6 = []  
list\_m7 = []  
list\_m8 = []  
list\_m9 = []  
list\_m10 = []  
list\_m11 = []  
list\_m12 = []

# Storing all ASCII Values  
list\_holder2 = [list\_m1, list\_m2, list\_m3, list\_m4, list\_m5, list\_m6, list\_m7, list\_m8, list\_m9, list\_m10, list\_m11, list\_m12]

Converts each binary value in list\_holder3 to ASCII and store it in each list in list\_holder2.

# Binary to ASCII  
for k in range(0, len(list\_holder3)):  
 # Converting each Binary in list\_holder3 to ASCII and storing in list\_holder2  
 list\_holder2[i] = ascii\_convertor\_func(list\_holder3[i])

Each element in the list\_holder2 is copied to their respective individual lists. And later ASCII values in each list are printed.

#Each element in the list\_holder2 is copied to their respective individual lists  
list\_m1 = list\_holder2[0]  
list\_m2 = list\_holder2[1]  
list\_m3 = list\_holder2[2]  
list\_m4 = list\_holder2[3]  
list\_m5 = list\_holder2[4]  
list\_m6 = list\_holder2[5]  
list\_m7 = list\_holder2[6]  
list\_m8 = list\_holder2[7]  
list\_m9 = list\_holder2[8]  
list\_m10 = list\_holder2[9]  
list\_m11 = list\_holder2[10]  
list\_m12 = list\_holder2[11]  
  
# Printing ASCII  
print(“ASCII for each Message is as follows:\n”)  
print(“Message 1: “, list\_m1,”\n”)  
print(“Message 2:”, list\_m2,”\n”)  
print(“Message 3:”, list\_m3,”\n”)  
print(“Message 4:”, list\_m4,”\n”)  
print(“Message 5:”, list\_m5,”\n”)  
print(“Message 6:”, list\_m6,”\n”)  
print(“Message 7:”, list\_m7,”\n”)  
print(“Message 8:”, list\_m8,”\n”)  
print(“Message 9:”, list\_m9,”\n”)  
print(“Message 10: “, list\_m10,”\n”)  
print(“Message 11: “, list\_m11,”\n”)  
print(“Message 12: “, list\_m12,”\n”)

### **Output:**

ASCII for each Message is as follows:

Message 1: [‘\x1b’, ‘\x06’, ‘\x00’, ‘\x00’, ‘\x1a’, ‘\x0e’, ‘\x11’, ‘\x04’, ‘\r’, ‘N’, ‘\x0c’, ‘S’, ‘C’, ‘\x17’, ‘\x0b’, ‘R’, ‘\x1d’, ‘O’, ‘D’, ‘\x1d’, ‘\x0e’, ‘\x03’, ‘E’, ‘\x05’, ‘\n’, ‘\x1c’, ‘\x1f’, ‘\x00’, ‘\x01’, ‘\t’, ‘W’, ‘F’, ‘/’, ‘\t’, ‘\x06’, ‘Y’, ‘M’, ‘\x0e’, ‘\x1c’, ‘D’, ‘\x01’, ‘\x01’, ‘\r’, ‘\x16’, ‘\n’, ‘\x00’, ‘\x04’, ‘\x0b’, ‘N’, ‘H’, “‘“, ‘\x1c’, ‘\x00’, ‘\x00’, ‘\x0e’, ‘\x11’, ‘\x1c’, ‘\x08’, ‘\x02’, ‘E’, ‘H’, ‘\x02’, ‘\x0f’, ‘\n’, ‘S’, ‘D’, ‘\x1b’, ‘H’, ‘\x11’, ‘H’, ‘\x15’, ‘O’, ‘\x00’, ‘\n’, ‘\x00’, ‘K’, ‘T’, ‘\x00’, ‘F’, ‘A’, ‘\r’, ‘\x01’, ‘\t’, ‘I’, ‘\x16’, ‘\x10’, ‘T’, ‘\x13’, ‘\r’, ‘\x03’, ‘R’, ‘\x02’, ‘\x06’, ‘\x1c’, ‘\x0c’, ‘\x01’, ‘Y’, ‘O’, ‘\x0b’, ‘A’, ‘\x06’]

Message 2: [‘\x0e’, ‘\x02’, ‘\t’, ‘\x00’, ‘\x02’, ‘\x07’, ‘C’, ‘\x1c’, ‘\x01’, ‘\x0b’, ‘E’, ‘B’, ‘\x0c’, ‘\x0e’, ‘\x05’, ‘S’, ‘D’, ‘\x06’, ‘N’, ‘T’, ‘\x1c’, ‘\r’, ‘E’, ‘W’, ‘\x18’, ‘\x1d’, ‘\x19’, ‘L’, ‘\x0b’, ‘F’, ‘C’, ‘\t’, ‘\x07’, ‘\x12’, ‘\x15’, ‘\x10’, ‘N’, ‘O’, ‘\x1c’, ‘\x0b’, ‘I’, ‘\x03’, ‘\x0e’, ‘\x00’, ‘\x1c’, ‘\x00’, ‘\x04’, ‘\x0b’, ‘\x08’, ‘O’, ‘<‘, ‘\x02’, ‘A’, ‘\x19’, ‘\x08’, ‘\x0c’, ‘\x06’, ‘I’, ‘\x1a’, ‘\r’, ‘A’, ‘\r’, ‘A’, ‘\x07’, ‘S’, ‘D’, ‘\r’, ‘R’, ‘\x1b’, ‘\t’, ‘\x01’, ‘C’, ‘\x16’, ‘\x1c’, ‘\x06’, ‘K’, ‘A’, ‘\x1c’, ‘F’, ‘V’, ‘\x06’, ‘\n’, ‘\x00’, ‘O’, ‘E’, ‘\x11’, ‘\x1a’, ‘R’, ‘\x00’, ‘O’, ‘\x01’, ‘\r’, ‘\x07’, ‘\t’, ‘\r’, ‘\x17’, ‘Y’, ‘L’, ‘\x0c’, ‘\x13’, ‘\t’]

Message 3: [‘\x1b’, ‘\x0b’, ‘\x06’, ‘H’, ‘\x03’, ‘\x0e’, ‘\x0f’, ‘\x07’, ‘\x0e’, ‘\x07’, ‘\x06’, ‘A’, ‘\x0f’, ‘A’, ‘\x1e’, ‘R’, ‘\x0b’, ‘\x08’, ‘R’, ‘\x11’, ‘\x1b’, ‘\x16’, ‘\x00’, ‘\x1f’, ‘\x0e’, ‘\x01’, ‘K’, ‘M’, ‘\n’, ‘\x14’, ‘E’, ‘\n’, ‘\x10’, ‘F’, ‘\x04’, ‘\x0b’, ‘O’, ‘\x19’, ‘\x1b’, ‘\x00’, ‘\x0c’, ‘\n’, ‘A’, ‘\x07’, ‘\n’, ‘\x00’, ‘\x1a’, ‘\x0c’, ‘\x1a’, ‘H’, ‘n’, ‘\x02’, ‘O’, ‘\x1f’, ‘\x04’, ‘C’, ‘\r’, ‘\x0f’, ‘\x08’, ‘\x0c’, ‘C’, ‘\n’, ‘\x04’, ‘\x00’, ‘T’, ‘D’, ‘\x02’, ‘E’, ‘\x15’, ‘\x06’, ‘\x16’, ‘\x00’, ‘\x11’, ‘\x00’, ‘\x00’, ‘K’, ‘G’, ‘\x00’, ‘\x0f’, ‘N’, ‘\x08’, ‘N’, ‘\x07’, ‘A’, ‘\x06’, ‘\x13’, ‘\x03’, ‘\x13’, ‘\x13’, ‘\x0b’, ‘\x01’, ‘\r’, ‘\x07’, ‘\t’, ‘\r’, ‘\x17’, ‘Y’, ‘L’, ‘\x0c’, ‘\x13’, ‘\t’]

Message 4: [‘\x1b’, ‘\x06’, ‘\x00’, ‘\x00’, ‘$’, ‘\x0f’, ‘\x17’, ‘\r’, ‘\x1b’, ‘\x00’, ‘\x00’, ‘T’, ‘C’, ‘\x08’, ‘\x1d’, ‘\x00’, ‘\x10’, ‘\x07’, ‘E’, ‘T’, ‘\x05’, ‘\n’, ‘S’, ‘\x03’, ‘O’, ‘\x1b’, ‘\x06’, ‘P’, ‘\x00’, ‘\x14’, ‘T’, ‘\x07’, ‘\x07’, ‘\x12’, ‘T’, ‘\n’, ‘I’, ‘\x01’, ‘\x15’, ‘\x08’, ‘\x0c’, ‘N’, ‘\x05’, ‘\x17’, ‘\x0f’, ‘E’, ‘\x01’, ‘\n’, ‘\x1e’, ‘M’, ‘+’, ‘\x01’, ‘T’, ‘M’, ‘\x08’, ‘\r’, ‘H’, ‘\x1d’, ‘\x06’, ‘\x00’, ‘\x00’, ‘\x0b’, ‘\x08’, ‘\x1d’, ‘T’, ‘\x0b’, ‘\x1d’, ‘Y’, ‘T’, ‘\x07’, ‘\x03’, ‘\x00’, ‘\x1f’, ‘\x1a’, ‘\x1f’, ‘\n’, ‘N’, ‘O’, ‘\x05’, ‘O’, ‘\x02’, ‘\x03’, ‘\x10’, ‘N’, ‘\x0c’, ‘\x1b’, ‘\x15’, ‘\x06’, ‘\x08’, ‘\x00’, ‘\x1c’, ‘D’, ‘\x1a’, ‘\x07’, ‘\x0f’, ‘\x11’, ‘\x1c’, ‘\x00’, ‘\x19’, ‘\t’, ‘\x0b’]

Message 5: [‘\x06’, ‘N’, ‘\x04’, ‘M’, ‘M’, ‘\x12’, ‘\x0c’, ‘\x1a’, ‘\x1b’, ‘\x17’, ‘E’, ‘T’, ‘\x0c’, ‘A’, ‘\x1d’, ‘A’, ‘\x1d’, ‘O’, ‘T’, ‘\x1c’, ‘\t’, ‘\x11’, ‘\x00’, ‘\x03’, ‘\x07’, ‘\x17’, ‘\x19’, ‘E’, ‘O’, ‘\x0f’, ‘S’, ‘F’, ‘\x1d’, ‘\t’, ‘\x1b’, ‘Y’, ‘M’, ‘\x1a’, ‘\x11’, ‘\x0c’, ‘I’, ‘\x1e’, ‘\x0e’, ‘\x1b’, ‘\x17’, ‘T’, ‘M’, ‘\x11’, ‘\x01’, ‘\x00’, ‘:’, ‘\x07’, ‘E’, ‘M’, ‘\x16’, ‘\n’, ‘\x1b’, ‘\x0c’, ‘\r’, ‘\x17’, ‘A’, ‘\x00’, ‘\n’, ‘N’, ‘T’, ‘\x0c’, ‘\x0e’, ‘T’, ‘T’, ‘\x04’, ‘\x0c’, ‘F’, ‘\x12’, ‘O’, ‘\x1b’, ‘\x18’, ‘\x00’, ‘\n’, ‘\x1e’, ‘T’, ‘\x06’, ‘\x00’, ‘\x06’, ‘T’, ‘E’, ‘\x17’, ‘\x1a’, ‘R’, ‘\x0e’, ‘\x1b’, ‘\x1a’, ‘\x01’, ‘\x1b’, ‘N’, ‘\x11’, ‘\x1e’, ‘\x18’, ‘N’, ‘\x08’, ‘\x15’, ‘\x1d’]

Message 6: [‘\x0c’, ‘\x06’, ‘\x04’, ‘M’, ‘\x1d’, ‘\x00’, ‘\x04’, ‘\x06’, ‘\x0c’, ‘N’, ‘\x0c’, ‘F’, ‘C’, ‘\x18’, ‘\x01’, ‘U’, ‘D’, ‘\x0e’, ‘R’, ‘\x11’, ‘H’, ‘\x16’, ‘E’, ‘\x12’, ‘\x04’, ‘\x1b’, ‘\x05’, ‘G’, ‘O’, ‘\x12’, ‘H’, ‘\x03’, ‘I’, ‘\x12’, ‘\x06’, ‘\x0c’, ‘T’, ‘\x07’, ‘R’, ‘\r’, ‘\x1a’, ‘N’, ‘\x03’, ‘\x17’, ‘\r’, ‘T’, ‘\x08’, ‘\x17’, ‘N’, ‘T’, ‘&’, ‘\x0e’, ‘N’, ‘M’, ‘\x00’, ‘C’, ‘\x04’, ‘\x00’, ‘\x0b’, ‘E’, ‘D’, ‘\x06’, ‘\x15’, ‘\x0b’, ‘C’, ‘\x10’, ‘\x00’, ‘R’, ‘T’, ‘!’, ‘\x11’, ‘\x00’, ‘\x12’, ‘\x01’, ‘\x11’, ‘\x04’, ‘U’, ‘\x1d’, ‘\x07’, ‘G’, ‘\n’, ‘\x1d’, ‘E’, ‘A’, ‘E’, ‘\x15’, ‘\x15’, ‘\x1c’, ‘A’, ‘\x1b’, ‘\x1d’, ‘D’, ‘\x0b’, ‘\x0b’, ‘A’, ‘\x17’, ‘\x01’, ‘P’, ‘\x0c’, ‘\x0f’, ‘\x1d’]

Message 7: [‘\r’, ‘\x1b’, ‘\x0c’, ‘L’, ‘\t’, ‘\x08’, ‘\r’, ‘\x0f’, ‘I’, ‘\x1a’, ‘\x00’, ‘C’, ‘\x0b’, ‘\x0f’, ‘\x07’, ‘C’, ‘\x05’, ‘\x03’, ‘\x00’, ‘\x07’, ‘\x11’, ‘\x16’, ‘T’, ‘\x12’, ‘\x02’, ‘\x01’, ‘K’, ‘I’, ‘\x01’, ‘\x10’, ‘O’, ‘\n’, ‘\x1f’, ‘\x03’, ‘\x07’, ‘Y’, ‘A’, ‘O’, ‘\x1e’, ‘\x0b’, ‘\x1d’, ‘N’, ‘\x0e’, ‘\x14’, ‘Y’, ‘H’, ‘\x0c’, ‘\x17’, ‘\n’, ‘\x00’, ‘9’, ‘\x00’, ‘R’, ‘\x06’, ‘A’, ‘\x02’, ‘\x06’, ‘\r’, ‘N’, ‘\x16’, ‘P’, ‘\x06’, ‘\x02’, ‘\x07’, ‘A’, ‘\x08’, ‘\x06’, ‘Z’, ‘\x11’, ‘\x0c’, ‘E’, ‘K’, ‘\x19’, ‘\x00’, ‘\x05’, ‘\x07’, ‘E’, ‘\x0b’, ‘\x01’, ‘E’, ‘O’, ‘\x02’, ‘\x04’, ‘N’, ‘\x02’, ‘\r’, ‘\x15’, ‘\x15’, ‘\x04’, ‘\x1c’, ‘R’, ‘\x05’, ‘\x07’, ‘\n’, ‘A’, ‘\x02’, ‘\x0b’, ‘O’, ‘\x19’, ‘\x0e’, ‘\r’]

Message 8: [‘\x06’, ‘\x1a’, ‘\x16’, ‘\x00’, ‘\x04’, ‘\x0c’, ‘\x13’, ‘\x07’, ‘\x1a’, ‘\x1d’, ‘\x0c’, ‘B’, ‘\x0f’, ‘\x04’, ‘N’, ‘T’, ‘\x0b’, ‘O’, ‘M’, ‘\x1b’, ‘\x1e’, ‘\x00’, ‘\x00’, ‘\x03’, ‘\x00’, ‘R’, ‘\x07’, ‘I’, ‘\x19’, ‘\x03’, ‘\x00’, ‘\x12’, ‘\x06’, ‘F’, ‘\x1b’, ‘\t’, ‘E’, ‘\x1d’, ‘\x13’, ‘\x10’, ‘\x0c’, ‘N’, ‘\x00’, ‘\x06’, ‘Y’, ‘A’, ‘\x03’, ‘\x1c’, ‘N’, ‘L’, ‘+’, ‘\x19’, ‘E’, ‘\x01’, ‘A’, ‘\x14’, ‘\x01’, ‘\x1d’, ‘\x06’, ‘\n’, ‘U’, ‘\x17’, ‘A’, ‘\x02’, ‘E’, ‘\x05’, ‘\x19’, ‘I’, ‘\x1a’, ‘\x0f’, ‘E’, ‘T’, ‘\x05’, ‘\x0e’, ‘\x11’, ‘\x0e’, ‘S’, ‘O’, ‘\x04’, ‘I’, ‘\x1b’, ‘\x1d’, ‘E’, ‘S’, ‘\x00’, ‘\x1d’, ‘\x19’, ‘\x1b’, ‘\x0f’, ‘\x08’, ‘\x1e’, ‘\x1d’, ‘I’, ‘\x03’, ‘\x04’, ‘\x13’, ‘\x17’, ‘I’, ‘\x03’, ‘\x06’, ‘\x02’]

Message 9: [‘\x1b’, ‘\x06’, ‘\x00’, ‘\x00’, ‘:’, ‘\x04’, ‘\x01’, ‘H’, ‘\x08’, ‘\x1d’, ‘E’, ‘i’, ‘C’, ‘\x04’, ‘\x00’, ‘V’, ‘\r’, ‘\x1c’, ‘A’, ‘\x13’, ‘\r’, ‘\x01’, ‘\x00’, ‘\x1e’, ‘\x1b’, ‘R’, ‘\x1c’, ‘E’, ‘O’, ‘\x0e’, ‘A’, ‘\x10’, ‘\x0c’, ‘F’, ‘\x1a’, ‘\x16’, ‘T’, ‘O’, ‘\x01’, ‘\x01’, ‘\x0c’, ‘\x00’, ‘A’, ‘\x1b’, ‘\r’, ‘\x00’, ‘\x14’, ‘\x00’, ‘\x1a’, ‘\x00’, ‘\x1a’, ‘\x07’, ‘E’, ‘M’, ‘\x07’, ‘\x16’, ‘\x1c’, ‘\x1c’, ‘\x1c’, ‘\x00’, ‘\x00’, ‘\n’, ‘\x12’, ‘N’, ‘S’, ‘\x10’, ‘\x06’, ‘L’, ‘\x18’, ‘H’, ‘\x16’, ‘O’, ‘W’, ‘\x02’, ‘\x07’, ‘\x08’, ‘H’, ‘O’, ‘\x04’, ‘I’, ‘\x08’, ‘\t’, ‘\x00’, ‘R’, ‘E’, ‘\x0c’, ‘\x1c’, ‘\x13’, ‘\x0f’, ‘O’, ‘\x06’, ‘\x0c’, ‘\x0c’, ‘N’, ‘\x11’, ‘\x13’, ‘\n’, ‘T’, ‘M’, ‘5’, ‘\x07’]

Message 10: [‘\x06’, ‘N’, ‘\x10’, ‘S’, ‘\x08’, ‘\x05’, ‘C’, ‘\x1c’, ‘\x06’, ‘N’, ‘\x11’, ‘H’, ‘\n’, ‘\x0f’, ‘\x05’, ‘\x00’, ‘\x10’, ‘\x07’, ‘A’, ‘\x00’, ‘H’, ‘\x06’, ‘Y’, ‘\x15’, ‘\n’, ‘\x00’, ‘\x18’, ‘P’, ‘\x0e’, ‘\x05’, ‘E’, ‘F’, ‘\x1e’, ‘\x07’, ‘\x07’, ‘Y’, ‘F’, ‘\x06’, ‘\x14’, ‘\x10’, ‘\x10’, ‘N’, ‘\x18’, ‘\x17’, ‘\x18’, ‘R’, ‘\x1e’, ‘E’, ‘\x0f’, ‘W’, ‘/’, ‘\x16’, ‘\x00’, ‘:’, ‘\t’, ‘\x02’, ‘\x1c’, ‘I’, “‘“, ‘E’, ‘T’, ‘\x0b’, ‘\x0e’, ‘\x1b’, ‘G’, ‘\x0c’, ‘\x1b’, ‘\x00’, ‘\x03’, ‘\t’, ‘\x16’, ‘\x00’, ‘\x11’, ‘\x06’, ‘\x14’, ‘\x1f’, ‘Y’, ‘O’, ‘\x1f’, ‘E’, ‘\x0e’, ‘\x1c’, ‘\x16’, ‘\x00’, ‘\x04’, ‘\x0f’, ‘\x15’, ‘\x0b’, ‘A’, ‘\x18’, ‘\x13’, ‘\x17’, ‘I’, ‘\x01’, ‘\x0f’, ‘\x1e’, ‘\x00’, ‘\x00’, ‘\x19’, ‘\x04’, ‘\x00’]

Message 11: [‘\x18’, ‘\x0b’, ‘E’, ‘A’, ‘\x1f’, ‘\x04’, ‘C’, ‘\x1c’, ‘\x01’, ‘\x0b’, ‘E’, ‘C’, ‘\x0b’, ‘\x08’, ‘\x02’, ‘D’, ‘\x16’, ‘\n’, ‘N’, ‘T’, ‘\x07’, ‘\x03’, ‘\x00’, ‘\x16’, ‘O’, ‘\x06’, ‘\x0e’, ‘C’, ‘\x07’, ‘\x08’, ‘O’, ‘\n’, ‘\x06’, ‘\x01’, ‘\x1d’, ‘\x1a’, ‘A’, ‘\x03’, ‘R’, ‘\x05’, ‘\x0e’, ‘\x0b’, ‘A’, ‘%’, ‘\x1c’, ‘\x00’, ‘\x05’, ‘\x04’, ‘\x18’, ‘E’, ‘n’, ‘\t’, ‘O’, ‘\x18’, ‘\x0f’, ‘\x07’, ‘H’, ‘\x1a’, ‘\x1a’, ‘\x17’, ‘E’, ‘\x02’, ‘\x0c’, ‘\x02’, ‘I’, ‘\n’, ‘\n’, ‘D’, ‘T’, ‘\x1f’, ‘\x04’, ‘Y’, ‘\x04’, ‘O’, ‘\x1d’, ‘\r’, ‘\x00’, ‘\x0b’, ‘\t’, ‘I’, ‘\x01’, ‘\t’, ‘E’, ‘M’, ‘\x10’, ‘\x1b’, ‘\x1c’, ‘R’, ‘\x0e’, ‘\t’, ‘R’, ‘\x0b’, ‘\x1c’, ‘\x1c’, ‘A’, ‘\x00’, ‘\x16’, ‘U’, ‘\x19’, ‘\x08’, ‘\x00’]

Message 12: [‘\x1c’, ‘\x1a’, ‘\x1c’, ‘L’, ‘\x08’, ‘A’, ‘\x16’, ‘\x1b’, ‘\x0c’, ‘\n’, ‘E’, ‘T’, ‘\x0c’, ‘A’, ‘\x0c’, ‘E’, ‘D’, ‘\x0e’, ‘N’, ‘T’, ‘\x01’, ‘\x0b’, ‘T’, ‘\x12’, ‘\x1d’, ‘\x13’, ‘\x08’, ‘T’, ‘\x06’, ‘\t’, ‘N’, ‘F’, ‘\x0b’, ‘\x03’, ‘\x00’, ‘\x0e’, ‘E’, ‘\n’, ‘\x1c’, ‘D’, ‘\x1d’, ‘\x06’, ‘\x04’, ‘R’, ‘\x11’, ‘U’, ‘\x00’, ‘\x04’, ‘\x00’, ‘\x00’, ‘=‘, ‘\x00’, ‘U’, ‘\x01’, ‘A’, ‘\x02’, ‘\x06’, ‘\r’, ‘N’, ‘\x11’, ‘O’, ‘\x0c’, ‘\r’, ‘\x1d’, ‘\x00’, ‘\x10’, ‘\x07’, ‘A’, ‘\x00’, ‘H’, ‘\x12’, ‘E’, ‘\x05’, ‘\n’, ‘R’, ‘\x07’, ‘I’, ‘\x02’, ‘\x0f’, ‘T’, ‘\x06’, ‘\x00’, ‘\x02’, ‘\x00’, ‘,’, ‘\x16’, ‘T’, ‘\x06’, ‘\t’, ‘\n’, ‘R’, ‘\x00’, ‘\x00’, ‘\t’, ‘\x08’, ‘\x06’, ‘\x18’, ‘L’, ‘M’, ‘\x04’, ‘\x1c’]

## **4) Find Key Values**

We will iterate through ASCII values stored in the list, row by row to look for letters in a key. If the row has mixed alphabets, then it’s a space and the space is stored in the key list. If the row has only one alphabet or the same alphabet repeating, then that alphabet is part of the key, and it is stored in a key list.

**Python Code Snippet with Explanation:**

store\_val is a list that will contain ASCII values of all lists per iteration. key\_1 is a list for storing alphabets after filtering out symbols. real\_key is a list for storing the actual key values.

# Key Finder program starts here  
store\_val = []  
key\_1 = [] #For storing alphabets after filtering out symbols   
real\_key = [] #For storing the actual key values  
  
# Alphabet list  
alphabet\_list = [“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”,  
 “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”]

It will iterate through each index value of different lists at a time and store it in the store\_val list in lower cases. Then it will delete any duplicate values. In the next step, it checks if the store\_val list has alphabets by comparing it with alphabet\_list which is a premade list of alphabets and if alphabets are found in the store\_val list then stores that alphabet in a new list called key\_1. To find out if there is space in the key, it will check if the length of the key is more than 1. If it’s more than 1 then there is space as we had removed duplicate alphabets in the previous step. So, if the list has more than 1 alphabet then it means space and the word space will be appended to the real\_key list. If there is only one alphabet, then that alphabet is stored in the real\_key list. After each iteration, it will clear the values in the store\_val list and key\_1 list to store new values.

for i in range(0, 101): # the number of rows it needs to iterate  
 store\_val.append(list\_m1[i].lower())  
 store\_val.append(list\_m2[i].lower())  
 store\_val.append(list\_m3[i].lower())  
 store\_val.append(list\_m4[i].lower())  
 store\_val.append(list\_m5[i].lower())  
 store\_val.append(list\_m6[i].lower())  
 store\_val.append(list\_m7[i].lower())  
 store\_val.append(list\_m8[i].lower())  
 store\_val.append(list\_m9[i].lower())  
 store\_val.append(list\_m10[i].lower())  
 store\_val.append(list\_m11[i].lower())  
 store\_val.append(list\_m12[i].lower())  
 store\_val = list(dict.fromkeys(store\_val)) # removes duplicates  
  
 for j in range(0, len(store\_val)):  
 if store\_val[j] in alphabet\_list: # Checking if the store\_val List has Alphabets  
 key\_1.append(store\_val[j]) # If the List has Alphabets then Append it to the key\_1  
  
 if len(key\_1) == 1: # If the key\_1 List has only 1 Alphabet then Append it to real\_key list  
 real\_key.append(key\_1[0])  
 else:  
 real\_key.append(“Space”) # If it has more than 1 alphabet then print Space  
  
 store\_val.clear() # Clearing the values stored in the List before moving to the next row  
 key\_1.clear() #Clearing values stored in key\_1 list before moving to the next row

This is to print the key/secret message in the real\_key list.

# Printing the real\_key/secret Message  
print()  
print(“The Key/Secret Message is as follows: “)  
print(real\_key,”\n”)  
print(“The Key/Secret Message Vertically: “)  
for i in range(0, len(real\_key)):  
 print(i, real\_key[i])

### **Output:**

The Key/Secret Message is as follows:

[‘Space’, ‘n’, ‘e’, ‘Space’, ‘m’, ‘a’, ‘c’, ‘h’, ‘i’, ‘n’, ‘e’, ‘Space’, ‘c’, ‘a’, ‘n’, ‘Space’, ‘d’, ‘o’, ‘Space’, ‘t’, ‘h’, ‘Space’, ‘Space’, ‘w’, ‘o’, ‘r’, ‘k’, ‘Space’, ‘o’, ‘f’, ‘Space’, ‘f’, ‘i’, ‘f’, ‘t’, ‘y’, ‘Space’, ‘o’, ‘r’, ‘d’, ‘i’, ‘n’, ‘a’, ‘r’, ‘y’, ‘Space’, ‘m’, ‘e’, ‘n’, ‘Space’, ‘n’, ‘Space’, ‘Space’, ‘m’, ‘a’, ‘c’, ‘h’, ‘i’, ‘n’, ‘e’, ‘Space’, ‘Space’, ‘a’, ‘n’, ‘Space’, ‘d’, ‘Space’, ‘Space’, ‘t’, ‘h’, ‘e’, ‘Space’, ‘w’, ‘o’, ‘r’, ‘k’, ‘Space’, ‘o’, ‘f’, ‘Space’, ‘o’, ‘n’, ‘e’, ‘Space’, ‘e’, ‘Space’, ‘t’, ‘r’, ‘a’, ‘o’, ‘r’, ‘d’, ‘i’, ‘n’, ‘a’, ‘Space’, ‘y’, ‘Space’, ‘m’, ‘a’, ‘Space’]

The Key/Secret Message Vertically:

0 Space

1 n

2 e

3 Space

4 m

5 a

6 c

7 h

8 i

9 n

10 e

11 Space

12 c

13 a

14 n

15 Space

16 d

17 o

18 Space

19 t

20 h

21 Space

22 Space

23 w

24 o

25 r

26 k

27 Space

28 o

29 f

30 Space

31 f

32 i

33 f

34 t

35 y

36 Space

37 o

38 r

39 d

40 i

41 n

42 a

43 r

44 y

45 Space

46 m

47 e

48 n

49 Space

50 n

51 Space

52 Space

53 m

54 a

55 c

56 h

57 i

58 n

59 e

60 Space

61 Space

62 a

63 n

64 Space

65 d

66 Space

67 Space

68 t

69 h

70 e

71 Space

72 w

73 o

74 r

75 k

76 Space

77 o

78 f

79 Space

80 o

81 n

82 e

83 Space

84 e

85 Space

86 t

87 r

88 a

89 o

90 r

91 d

92 i

93 n

94 a

95 Space

96 y

97 Space

98 m

99 a

100 Space

## **5) Printing The Complete Secret Message**

After looking at the key we got from the program, we understood the complete secret message. The secret message is as follows: **One machine can do the work of fifty ordinary men; no machine can do the work of one extraordinary man.**

**Python Code Snippet with Explanation:**

# Printing the complete Secret Message  
print()  
print(“Secret Message is as follows: “)  
print(“One machine can do the work of fifty ordinary men; no machine can do the work of one extraordinary man”)

### **Output:**

The secret Message is as follows:

One machine can do the work of fifty ordinary men; no machine can do the work of one extraordinary man

# **Full Python Code:**

# CSCI368 Assignment 1 Code  
  
# Functions start here  
  
# Function to find XOR values, it returns the result in Hex String  
def xor\_func(m1, m13):  
 xor\_hex\_value = hex(m1 ^ m13)  
 return xor\_hex\_value  
  
# Function to convert the above-found Hex String to Binary  
def hex\_to\_binary\_func(xor\_hex\_value):  
 hex1 = xor\_hex\_value[2:]  
 binary\_value = bin(int(‘1’ + hex1, 16))[3:]  
 return binary\_value  
  
# Function to convert the above Binary to ASCII  
def toString(binaryString):  
 return ““.join([chr(int(binaryString[i:i + 8], 2)) for i in range(0, len(binaryString), 8)])  
  
# Function to convert 8 bits at a time to ASCII and store it  
def ascii\_convertor\_func(binary\_value):  
 mystring = ““  
 store\_value = ““  
 list1 = []  
 i = 0  
 # print(“Printing 8 bits of Binary:”)  
  
 for x in binary\_value:  
  
 mystring = mystring + x  
  
 # Converting 8 bits to ASCII  
 if len(mystring) >= 8:  
 store\_value = toString(mystring)  
  
 list1.append(store\_value)  
  
 i = i + 1  
  
 mystring = ““  
  
 return list1  
  
# Function to print the stored ASCII Values  
def print\_ascii\_func(list0):  
 print(“Printing the stored ASCII Values”)  
 i = 0  
 for x in list0:  
 i = i + 1  
 print(i, x)  
  
# End of Functions  
  
# Main Program starts here  
  
# Storing All Messages  
  
# We trimmed Messages 1 to 12 to match the length of Message 13  
m1 = 0x0f351c71e76f5fbe548d4c54a69a2bcb3d2e4ceb3cfb5250c24dff419949683f8ed3a5f04f57116fe797410d2c138b60db6da534a7abe0f658b65b5c0ccbeb67d1c9d9216d8befeb35173f3596f40d4fa84e1e702818fbc06bec90d4315fceacfa7112c3e5 #5d74aaf3394bb08f7504a8e5019c4e3e838e0f364946f31721a49ad2d24ff6775efcb4f79fe4217a01b43cb5068bf3b52ca76543187274  
m2 = 0x1a311571ff660da658c80545e98325ca646746a22ef55202d04cf90d93067c70a6c8b6b94c161120af95421b3a138b609d6abe2ae6b2e6eb42f7431405c4a56ad1c9cf3b67cafbe72301393583e80d58a34517767b19b58166a0c3db304acfbafa721591ea #4d398af07c49b69a7118fcff4889597aca81433b4953f50b2bbbdf9dcd0aff7013d3b5a3d3ec2b73019c3aa91b8bddf411a72b480c636cc6597494151386  
m3 = 0x0f381a39fe6f41bd57c44646eacc3ecb2b695ae729ee174ac650ab0c92547a73b19ca7a24d40162bea9c0d1c2c1395678f6dec2ae8b4eaa449b1511507c3e06dd6c9c02c69c5eca4241d3f3585f44440ad011078381bacc075e4c3db304acfbafa721591ea  
m4 = 0x0f351c71d96e59b742c34053a6853d9930664da237f24456874ae61198546b7ea6c8f7a34b581823ead8490c29568e618b68a929f3e6e6ea0ca35f1944c2ec70d686df3028c4f9a42a0720748cbb4e41a74c07773213bad56eefde922d44cdbcbf3e008be8 #0870a5eb7c55ab907f18fcf347dd433bcf83432d0849e80c22b0  
m5 = 0x127d183cb07342a042d40553e9cc3dd83d2e5cea3be91756cf46f904d74f6c3fbcd3b8f04f431c27af8842003147c27a9425b82fe2e6f8ed5fb2540e05c9ee23d681cc3d28c7f6e227522466c2fe555aa34f116d7b1fb58168f4d8d72c0dd3b3bb701197fe #087baefe784eac9c3002b4f9488f0029c08606341d49ef113ff7cdd8c60abe6f5cefbff792e9317e4f9d36b948dfddf409e264580f65  
m6 = 0x1835183ce0614abc558d4c41a69521cc646f5ae77aee5247cc4ae506d752777ae8c8a5a5565e5f26fcd84f0c2b47877cdb71a426e9e6eea440be525c00cff166c19dc23b28e2eba4271c2e7a97e94c49af5252787b1dbacf27f4df923c4883baa26e158dfe #416faebd7c4dba973004b9ff4a914529d0cf1432004cf94520bedf9dd00aea6750e9b5a580ad266d44de3cb304d295f447a1634c117a68c41e67d50d09c35928a62e6d6648371b40ac83b274cecb04d6c41b6fba  
m7 = 0x1928103df46943b510d94044ee8227da256208f123ee4347ca50ab0899507073bed9a4f043161320fbd8420f7f5b837c9f25bb28f5adafe542b3170f14cfe66ac385c4336dcfbfef2c1d3a7987ff4a4bea4d13773c05bac662f390d3304983afa871008cee #4775b8bd7a54bb907e11fcfd4f99003ec68d163d0e49f2026ca3dfcec006f06513fcb4b3d3ff2279409d27b21ac2dbf2  
m8 = 0x12290a71f96d5dbd43de4c45ea896ecd2b2e45ed2cf81756c803e70881433f6ba79cb8a047441e3bead84c1d7f528c77db69a931e2aaaff345a35f1311dea56fc788db2066ccbff030132e7091bb4f47be52526a3e15b6c869e7dccb7e40c6beb4771a84e1 #4d6ab8bd7f49be9e7d13b2e852dd4f3c839f06281a4ff20420f7d3d3d200ec6f52e9b3b89d  
m9 = 0x0f351c71c7654ff251de056ea68920cf2d7d49e53ff9174bd303fc04d74e7e69ad9cb9bf56160c2aea960d002b139b6b8f25982fe2e6e9f158a2451944c3f623d19dc425648beceb621f38768abb4f47ad46176b7b04b3c069a0c4da3b0dd3bea96a54b7e4 #453989f86b55ba8b635b90f944  
m10 = 0x127d0c22f5640da65f8d514fef822599306649f67afe4e40c251f81196457a3fbfdda4f0445f193bf6d8540c3e41912e9a72ad3ea791e7e558f77e5c10c2ea76c581d9697fcaeca4241b2b619bbb544bab5301393a07bad827f7d1c17e42cdb3a33e0086e3 #0860aefc6b48ff986717a5bc6093447ad487022e4969bc1124b8cfdadc1bbe7552eefaa396e36766449f21ae48cac2f41ee262595d616cd95963990b03824934ea2a287844722140b583a2638bcf16c3df0323a212740f3d5b517fbd10e4e253512e  
m11 = 0x0c385930e2650da658c80544ee8522dd366b46a235fb17438757ee029f487073a7dbbeb3435a5f2ee89d0d3e3a138a6f8d60ec21e8b3e1e00ca4430e01cbe86fcb87c82d28dcfefd31522273c2ff4247a44652742e13b38168e690dd2b5f83adb56b008ae3 #4d39bcf26b50ffa9621fb2e84893477aca9c43340600f00a22b0dfcf941bf66713f2b4bb8aad307e58de3cbb48d9d0e515ad6f581e7f63cd59609a160d900d1faf2329634f354814b793bc37c3d700d5c71271e30d740e7815516dbd1af8a344533fcb  
m12 = 0x0829003df52058a155c90553e9cc2cdc646f46a233f34347d542e8159e49713faad9a3a74753116ffb90484937468f6f9525bf28f2aaafe542b317080bc5e970829dc5287c8be8e130176d798bf6445aa34f1539121efbd56fe590d6374acaabbb725486ff #4939a2e9394cb6957c56b4fd5798002ecccf00350445bc033eb8d79dc007fb2240f2afbbd3ec2b704f9b  
  
m13 = 0x14331c71fd614eba59c34007e58d2099206108f632f81755c851e04198403f79a1daa3a902590d2be6964c1b26138f6b95258228a7abeee744be591944c9e46d828dc2697cc3faa4351d3f7ec2f44b0ea54f17393e08afd366efc2d63743c2ada33e1982e3  
  
list\_m1 = []  
list\_m2 = []  
list\_m3 = []  
list\_m4 = []  
list\_m5 = []  
list\_m6 = []  
list\_m7 = []  
list\_m8 = []  
list\_m9 = []  
list\_m10 = []  
list\_m11 = []  
list\_m12 = []  
  
# Storing All Messages  
list\_holder1 = [m1, m2, m3, m4, m5, m6, m7, m8, m9, m10, m11, m12]  
  
# Storing all ASCII Values  
list\_holder2 = [list\_m1, list\_m2, list\_m3, list\_m4, list\_m5, list\_m6, list\_m7, list\_m8, list\_m9, list\_m10, list\_m11, list\_m12]  
  
list\_holder3 = []  
  
for i in range(0, 12):  
 # Calling the XOR Function  
 xor\_hex\_value = xor\_func(list\_holder1[i], m13) ##List\_holder1 is a list that contains all messages and each messages XOR’d with message 13 using hex\_function and the value is stored in xor\_hex\_value  
 print(“Hex value of Message”, i + 1, “XOR’d with Message 13:”)  
 print(xor\_hex\_value + “\n”)  
  
 # Calling the Hex String to Binary Function  
 binary\_value = hex\_to\_binary\_func(xor\_hex\_value)  
 print(“Binary of Message”, i + 1, “XOR’d with message 13:”)  
  
 # Adding four 0’s to the Binary of XOR’d Hex’s of messages 1, 4, 5, 6, 7, 9  
 # because all these messages start with 1 instead of 0  
 if(i == 1 or i == 4 or i == 5 or i == 6 or i == 7 or i == 9):  
 binary\_value = “0000” + binary\_value  
 print(binary\_value + “\n”)  
 list\_holder3.append(binary\_value) #Storing binary values to list\_holder3  
  
 else:  
 print(binary\_value + “\n”)  
 list\_holder3.append(binary\_value) #Storing binary values to list\_holder3  
  
 print(“---------------------------------------------------------------------------------------------------------------” + “\n”)  
  
 # Binary to ASCII  
 for k in range(0, len(list\_holder3)):  
 # Converting each Binary in list\_holder3 to ASCII and storing in list\_holder2  
 list\_holder2[i] = ascii\_convertor\_func(list\_holder3[i])  
  
  
# Each element in the list\_holder2 is copied to their respective individual lists  
list\_m1 = list\_holder2[0]  
list\_m2 = list\_holder2[1]  
list\_m3 = list\_holder2[2]  
list\_m4 = list\_holder2[3]  
list\_m5 = list\_holder2[4]  
list\_m6 = list\_holder2[5]  
list\_m7 = list\_holder2[6]  
list\_m8 = list\_holder2[7]  
list\_m9 = list\_holder2[8]  
list\_m10 = list\_holder2[9]  
list\_m11 = list\_holder2[10]  
list\_m12 = list\_holder2[11]  
  
# Printing ASCII  
print(“ASCII for each Message is as follows:\n”)  
print(“Message 1: “, list\_m1,”\n”)  
print(“Message 2:”, list\_m2,”\n”)  
print(“Message 3:”, list\_m3,”\n”)  
print(“Message 4:”, list\_m4,”\n”)  
print(“Message 5:”, list\_m5,”\n”)  
print(“Message 6:”, list\_m6,”\n”)  
print(“Message 7:”, list\_m7,”\n”)  
print(“Message 8:”, list\_m8,”\n”)  
print(“Message 9:”, list\_m9,”\n”)  
print(“Message 10: “, list\_m10,”\n”)  
print(“Message 11: “, list\_m11,”\n”)  
print(“Message 12: “, list\_m12,”\n”)  
  
# Key Finder program starts here  
store\_val = []  
key\_1 = [] #For storing alphabets after filtering out symbols  
real\_key = [] #For storing the actual key values  
  
# Alphabet list  
alphabet\_list = [“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”,  
 “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”]  
  
  
  
# Finding key from ASCII  
for i in range(0, 101): # the number of rows it needs to iterate  
 store\_val.append(list\_m1[i].lower())  
 store\_val.append(list\_m2[i].lower())  
 store\_val.append(list\_m3[i].lower())  
 store\_val.append(list\_m4[i].lower())  
 store\_val.append(list\_m5[i].lower())  
 store\_val.append(list\_m6[i].lower())  
 store\_val.append(list\_m7[i].lower())  
 store\_val.append(list\_m8[i].lower())  
 store\_val.append(list\_m9[i].lower())  
 store\_val.append(list\_m10[i].lower())  
 store\_val.append(list\_m11[i].lower())  
 store\_val.append(list\_m12[i].lower())  
 store\_val = list(dict.fromkeys(store\_val)) # removes duplicates  
  
 for j in range(0, len(store\_val)):  
 if store\_val[j] in alphabet\_list: # Checking if the store\_val List has Alphabets  
 key\_1.append(store\_val[j]) # If the List has Alphabets then Append it to the key\_1  
  
 if len(key\_1) == 1: # If the key\_1 List has only 1 Alphabet then Append it to real\_key list  
 real\_key.append(key\_1[0])  
 else:  
 real\_key.append(“Space”) # If it has more than 1 alphabet then print Space  
  
 store\_val.clear() # Clearing the values stored in the List before moving to the next row  
 key\_1.clear() # Clearing values stored in key\_1 list before moving to the next row  
  
# Printing the real\_key/secret Message  
print()  
print(“The Key/Secret Message is as follows: “)  
print(real\_key,”\n”)  
print(“The Key/Secret Message Vertically: “)  
for i in range(0, len(real\_key)):  
 print(i, real\_key[i])  
  
# Printing the complete Secret Message  
print()  
print(“Secret Message is as follows: “)  
print(“One machine can do the work of fifty ordinary men; no machine can do the work of one extraordinary man”)

# **Complete Output**

Hex value of Message 1 XOR’d with Message 13:

0x1b0600001a0e11040d4e0c5343170b521d4f441d0e0345050a1c1f00010957462f0906594d0e1c4401010d160a00040b4e48271c00000e111c08024548020f0a53441b481148154f000a004b540046410d010949161054130d035202061c0c01594f0b4106

Binary of Message 1 XOR’d with message 13:

0001101100000110000000000000000000011010000011100001000100000100000011010100111000001100010100110100001100010111000010110101001000011101010011110100010000011101000011100000001101000101000001010000101000011100000111110000000000000001000010010101011101000110001011110000100100000110010110010100110100001110000111000100010000000001000000010000110100010110000010100000000000000100000010110100111001001000001001110001110000000000000000000000111000010001000111000000100000000010010001010100100000000010000011110000101001010011010001000001101101001000000100010100100000010101010011110000000000001010000000000100101101010100000000000100011001000001000011010000000100001001010010010001011000010000010101000001001100001101000000110101001000000010000001100001110000001100000000010101100101001111000010110100000100000110

---------------------------------------------------------------------------------------------------------------

Hex value of Message 2 XOR’d with Message 13:

0xe0209000207431c010b45420c0e055344064e541c0d4557181d194c0b464309071215104e4f1c0b49030e001c00040b084f3c024119080c06491a0d410d410753440d521b090143161c064b411c4656060a004f45111a52004f010d07090d17594c0c1309

Binary of Message 2 XOR’d with message 13:

0000111000000010000010010000000000000010000001110100001100011100000000010000101101000101010000100000110000001110000001010101001101000100000001100100111001010100000111000000110101000101010101110001100000011101000110010100110000001011010001100100001100001001000001110001001000010101000100000100111001001111000111000000101101001001000000110000111000000000000111000000000000000100000010110000100001001111001111000000001001000001000110010000100000001100000001100100100100011010000011010100000100001101010000010000011101010011010001000000110101010010000110110000100100000001010000110001011000011100000001100100101101000001000111000100011001010110000001100000101000000000010011110100010100010001000110100101001000000000010011110000000100001101000001110000100100001101000101110101100101001100000011000001001100001001

---------------------------------------------------------------------------------------------------------------

Hex value of Message 3 XOR’d with Message 13:

0x1b0b0648030e0f070e0706410f411e520b0852111b16001f0e014b4d0a14450a1046040b4f191b000c0a41070a001a0c1a486e024f1f04430d0f080c430a040054440245150616001100004b47000f4e084e074106130313130b010d07090d17594c0c1309

Binary of Message 3 XOR’d with message 13:

0001101100001011000001100100100000000011000011100000111100000111000011100000011100000110010000010000111101000001000111100101001000001011000010000101001000010001000110110001011000000000000111110000111000000001010010110100110100001010000101000100010100001010000100000100011000000100000010110100111100011001000110110000000000001100000010100100000100000111000010100000000000011010000011000001101001001000011011100000001001001111000111110000010001000011000011010000111100001000000011000100001100001010000001000000000001010100010001000000001001000101000101010000011000010110000000000001000100000000000000000100101101000111000000000000111101001110000010000100111000000111010000010000011000010011000000110001001100010011000010110000000100001101000001110000100100001101000101110101100101001100000011000001001100001001

---------------------------------------------------------------------------------------------------------------

Hex value of Message 4 XOR’d with Message 13:

0x1b060000240f170d1b00005443081d0010074554050a53034f1b0650001454070712540a490115080c4e05170f45010a1e4d2b01544d080d481d0600000b081d540b1d59540703001f1a1f0a4e4f054f0203104e0c1b150608001c441a070f111c0019090b

Binary of Message 4 XOR’d with message 13:

0001101100000110000000000000000000100100000011110001011100001101000110110000000000000000010101000100001100001000000111010000000000010000000001110100010101010100000001010000101001010011000000110100111100011011000001100101000000000000000101000101010000000111000001110001001001010100000010100100100100000001000101010000100000001100010011100000010100010111000011110100010100000001000010100001111001001101001010110000000101010100010011010000100000001101010010000001110100000110000000000000000000001011000010000001110101010100000010110001110101011001010101000000011100000011000000000001111100011010000111110000101001001110010011110000010101001111000000100000001100010000010011100000110000011011000101010000011000001000000000000001110001000100000110100000011100001111000100010001110000000000000110010000100100001011

---------------------------------------------------------------------------------------------------------------

Hex value of Message 5 XOR’d with Message 13:

0x64e044d4d120c1a1b1745540c411d411d4f541c09110003071719454f0f53461d091b594d1a110c491e0e1b17544d1101003a07454d160a1b0c0d1741000a4e540c0e5454040c46124f1b18000a1e540600065445171a520e1b1a011b4e111e184e08151d

Binary of Message 5 XOR’d with message 13:

0000011001001110000001000100110101001101000100100000110000011010000110110001011101000101010101000000110001000001000111010100000100011101010011110101010000011100000010010001000100000000000000110000011100010111000110010100010101001111000011110101001101000110000111010000100100011011010110010100110100011010000100010000110001001001000111100000111000011011000101110101010001001101000100010000000100000000001110100000011101000101010011010001011000001010000110110000110000001101000101110100000100000000000010100100111001010100000011000000111001010100010101000000010000001100010001100001001001001111000110110001100000000000000010100001111001010100000001100000000000000110010101000100010100010111000110100101001000001110000110110001101000000001000110110100111000010001000111100001100001001110000010000001010100011101

---------------------------------------------------------------------------------------------------------------

Hex value of Message 6 XOR’d with Message 13:

0xc06044d1d0004060c4e0c4643180155440e521148164512041b05474f1248034912060c5407520d1a4e03170d5408174e54260e4e4d004304000b454406150b431000525421110012011104551d07470a1d45414515151c411b1d440b0b411701500c0f1d

Binary of Message 6 XOR’d with message 13:

0000110000000110000001000100110100011101000000000000010000000110000011000100111000001100010001100100001100011000000000010101010101000100000011100101001000010001010010000001011001000101000100100000010000011011000001010100011101001111000100100100100000000011010010010001001000000110000011000101010000000111010100100000110100011010010011100000001100010111000011010101010000001000000101110100111001010100001001100000111001001110010011010000000001000011000001000000000000001011010001010100010000000110000101010000101101000011000100000000000001010010010101000010000100010001000000000001001000000001000100010000010001010101000111010000011101000111000010100001110101000101010000010100010100010101000101010001110001000001000110110001110101000100000010110000101101000001000101110000000101010000000011000000111100011101

---------------------------------------------------------------------------------------------------------------

Hex value of Message 7 XOR’d with Message 13:

0xd1b0c4c09080d0f491a00430b0f0743050300071116541202014b4901104f0a1f030759414f1e0b1d4e0e1459480c170a00390052064102060d4e16500602074108065a110c454b19000507450b01454f02044e020d1515041c5205070a41020b4f190e0d

Binary of Message 7 XOR’d with message 13:

0000110100011011000011000100110000001001000010000000110100001111010010010001101000000000010000110000101100001111000001110100001100000101000000110000000000000111000100010001011001010100000100100000001000000001010010110100100100000001000100000100111100001010000111110000001100000111010110010100000101001111000111100000101100011101010011100000111000010100010110010100100000001100000101110000101000000000001110010000000001010010000001100100000100000010000001100000110101001110000101100101000000000110000000100000011101000001000010000000011001011010000100010000110001000101010010110001100100000000000001010000011101000101000010110000000101000101010011110000001000000100010011100000001000001101000101010001010100000100000111000101001000000101000001110000101001000001000000100000101101001111000110010000111000001101

---------------------------------------------------------------------------------------------------------------

Hex value of Message 8 XOR’d with Message 13:

0x61a1600040c13071a1d0c420f044e540b4f4d1b1e000003005207491903001206461b09451d13100c4e00065941031c4e4c2b1945014114011d060a55174102450519491a0f4554050e110e534f04491b1d4553001d191b0f081e1d490304131749030602

Binary of Message 8 XOR’d with message 13:

0000011000011010000101100000000000000100000011000001001100000111000110100001110100001100010000100000111100000100010011100101010000001011010011110100110100011011000111100000000000000000000000110000000001010010000001110100100100011001000000110000000000010010000001100100011000011011000010010100010100011101000100110001000000001100010011100000000000000110010110010100000100000011000111000100111001001100001010110001100101000101000000010100000100010100000000010001110100000110000010100101010100010111010000010000001001000101000001010001100101001001000110100000111101000101010101000000010100001110000100010000111001010011010011110000010001001001000110110001110101000101010100110000000000011101000110010001101100001111000010000001111000011101010010010000001100000100000100110001011101001001000000110000011000000010

---------------------------------------------------------------------------------------------------------------

Hex value of Message 9 XOR’d with Message 13:

0x1b0600003a040148081d4569430400560d1c41130d01001e1b521c454f0e41100c461a16544f01010c00411b0d0014001a001a07454d07161c1c1c00000a124e5310064c1848164f57020708484f044908090052450c1c130f4f060c0c4e11130a544d3507

Binary of Message 9 XOR’d with message 13:

0001101100000110000000000000000000111010000001000000000101001000000010000001110101000101011010010100001100000100000000000101011000001101000111000100000100010011000011010000000100000000000111100001101101010010000111000100010101001111000011100100000100010000000011000100011000011010000101100101010001001111000000010000000100001100000000000100000100011011000011010000000000010100000000000001101000000000000110100000011101000101010011010000011100010110000111000001110000011100000000000000000000001010000100100100111001010011000100000000011001001100000110000100100000010110010011110101011100000010000001110000100001001000010011110000010001001001000010000000100100000000010100100100010100001100000111000001001100001111010011110000011000001100000011000100111000010001000100110000101001010100010011010011010100000111

---------------------------------------------------------------------------------------------------------------

Hex value of Message 10 XOR’d with Message 13:

0x64e10530805431c064e11480a0f050010074100480659150a0018500e0545461e07075946061410104e181718521e450f572f16003a09021c492745540b0e1b470c1b00030916001106141f594f1f450e1c1600040f150b4118131749010f1e0000190400

Binary of Message 10 XOR’d with message 13:

0000011001001110000100000101001100001000000001010100001100011100000001100100111000010001010010000000101000001111000001010000000000010000000001110100000100000000010010000000011001011001000101010000101000000000000110000101000000001110000001010100010101000110000111100000011100000111010110010100011000000110000101000001000000010000010011100001100000010111000110000101001000011110010001010000111101010111001011110001011000000000001110100000100100000010000111000100100100100111010001010101010000001011000011100001101101000111000011000001101100000000000000110000100100010110000000000001000100000110000101000001111101011001010011110001111101000101000011100001110000010110000000000000010000001111000101010000101101000001000110000001001100010111010010010000000100001111000111100000000000000000000110010000010000000000

---------------------------------------------------------------------------------------------------------------

Hex value of Message 11 XOR’d with Message 13:

0x180b45411f04431c010b45430b080244160a4e54070300164f060e4307084f0a06011d1a410352050e0b41251c00050418456e094f180f07481a1a1745020c02490a0a44541f0459044f1d0d000b09490109454d101b1c520e09520b1c1c41001655190800

Binary of Message 11 XOR’d with message 13:

0001100000001011010001010100000100011111000001000100001100011100000000010000101101000101010000110000101100001000000000100100010000010110000010100100111001010100000001110000001100000000000101100100111100000110000011100100001100000111000010000100111100001010000001100000000100011101000110100100000100000011010100100000010100001110000010110100000100100101000111000000000000000101000001000001100001000101011011100000100101001111000110000000111100000111010010000001101000011010000101110100010100000010000011000000001001001001000010100000101001000100010101000001111100000100010110010000010001001111000111010000110100000000000010110000100101001001000000010000100101000101010011010001000000011011000111000101001000001110000010010101001000001011000111000001110001000001000000000001011001010101000110010000100000000000

---------------------------------------------------------------------------------------------------------------

Hex value of Message 12 XOR’d with Message 13:

0x1c1a1c4c0841161b0c0a45540c410c45440e4e54010b54121d13085406094e460b03000e450a1c441d0604521155000400003d0055014102060d4e114f0c0d1d0010074100481245050a520749020f54060002002c165406090a520000090806184c4d041c

Binary of Message 12 XOR’d with message 13:

0001110000011010000111000100110000001000010000010001011000011011000011000000101001000101010101000000110001000001000011000100010101000100000011100100111001010100000000010000101101010100000100100001110100010011000010000101010000000110000010010100111001000110000010110000001100000000000011100100010100001010000111000100010000011101000001100000010001010010000100010101010100000000000001000000000000000000001111010000000001010101000000010100000100000010000001100000110101001110000100010100111100001100000011010001110100000000000100000000011101000001000000000100100000010010010001010000010100001010010100100000011101001001000000100000111101010100000001100000000000000010000000000010110000010110010101000000011000001001000010100101001000000000000000000000100100001000000001100001100001001100010011010000010000011100

---------------------------------------------------------------------------------------------------------------

ASCII for each Message is as follows:

Message 1: [‘\x1b’, ‘\x06’, ‘\x00’, ‘\x00’, ‘\x1a’, ‘\x0e’, ‘\x11’, ‘\x04’, ‘\r’, ‘N’, ‘\x0c’, ‘S’, ‘C’, ‘\x17’, ‘\x0b’, ‘R’, ‘\x1d’, ‘O’, ‘D’, ‘\x1d’, ‘\x0e’, ‘\x03’, ‘E’, ‘\x05’, ‘\n’, ‘\x1c’, ‘\x1f’, ‘\x00’, ‘\x01’, ‘\t’, ‘W’, ‘F’, ‘/’, ‘\t’, ‘\x06’, ‘Y’, ‘M’, ‘\x0e’, ‘\x1c’, ‘D’, ‘\x01’, ‘\x01’, ‘\r’, ‘\x16’, ‘\n’, ‘\x00’, ‘\x04’, ‘\x0b’, ‘N’, ‘H’, “‘“, ‘\x1c’, ‘\x00’, ‘\x00’, ‘\x0e’, ‘\x11’, ‘\x1c’, ‘\x08’, ‘\x02’, ‘E’, ‘H’, ‘\x02’, ‘\x0f’, ‘\n’, ‘S’, ‘D’, ‘\x1b’, ‘H’, ‘\x11’, ‘H’, ‘\x15’, ‘O’, ‘\x00’, ‘\n’, ‘\x00’, ‘K’, ‘T’, ‘\x00’, ‘F’, ‘A’, ‘\r’, ‘\x01’, ‘\t’, ‘I’, ‘\x16’, ‘\x10’, ‘T’, ‘\x13’, ‘\r’, ‘\x03’, ‘R’, ‘\x02’, ‘\x06’, ‘\x1c’, ‘\x0c’, ‘\x01’, ‘Y’, ‘O’, ‘\x0b’, ‘A’, ‘\x06’]

Message 2: [‘\x0e’, ‘\x02’, ‘\t’, ‘\x00’, ‘\x02’, ‘\x07’, ‘C’, ‘\x1c’, ‘\x01’, ‘\x0b’, ‘E’, ‘B’, ‘\x0c’, ‘\x0e’, ‘\x05’, ‘S’, ‘D’, ‘\x06’, ‘N’, ‘T’, ‘\x1c’, ‘\r’, ‘E’, ‘W’, ‘\x18’, ‘\x1d’, ‘\x19’, ‘L’, ‘\x0b’, ‘F’, ‘C’, ‘\t’, ‘\x07’, ‘\x12’, ‘\x15’, ‘\x10’, ‘N’, ‘O’, ‘\x1c’, ‘\x0b’, ‘I’, ‘\x03’, ‘\x0e’, ‘\x00’, ‘\x1c’, ‘\x00’, ‘\x04’, ‘\x0b’, ‘\x08’, ‘O’, ‘<‘, ‘\x02’, ‘A’, ‘\x19’, ‘\x08’, ‘\x0c’, ‘\x06’, ‘I’, ‘\x1a’, ‘\r’, ‘A’, ‘\r’, ‘A’, ‘\x07’, ‘S’, ‘D’, ‘\r’, ‘R’, ‘\x1b’, ‘\t’, ‘\x01’, ‘C’, ‘\x16’, ‘\x1c’, ‘\x06’, ‘K’, ‘A’, ‘\x1c’, ‘F’, ‘V’, ‘\x06’, ‘\n’, ‘\x00’, ‘O’, ‘E’, ‘\x11’, ‘\x1a’, ‘R’, ‘\x00’, ‘O’, ‘\x01’, ‘\r’, ‘\x07’, ‘\t’, ‘\r’, ‘\x17’, ‘Y’, ‘L’, ‘\x0c’, ‘\x13’, ‘\t’]

Message 3: [‘\x1b’, ‘\x0b’, ‘\x06’, ‘H’, ‘\x03’, ‘\x0e’, ‘\x0f’, ‘\x07’, ‘\x0e’, ‘\x07’, ‘\x06’, ‘A’, ‘\x0f’, ‘A’, ‘\x1e’, ‘R’, ‘\x0b’, ‘\x08’, ‘R’, ‘\x11’, ‘\x1b’, ‘\x16’, ‘\x00’, ‘\x1f’, ‘\x0e’, ‘\x01’, ‘K’, ‘M’, ‘\n’, ‘\x14’, ‘E’, ‘\n’, ‘\x10’, ‘F’, ‘\x04’, ‘\x0b’, ‘O’, ‘\x19’, ‘\x1b’, ‘\x00’, ‘\x0c’, ‘\n’, ‘A’, ‘\x07’, ‘\n’, ‘\x00’, ‘\x1a’, ‘\x0c’, ‘\x1a’, ‘H’, ‘n’, ‘\x02’, ‘O’, ‘\x1f’, ‘\x04’, ‘C’, ‘\r’, ‘\x0f’, ‘\x08’, ‘\x0c’, ‘C’, ‘\n’, ‘\x04’, ‘\x00’, ‘T’, ‘D’, ‘\x02’, ‘E’, ‘\x15’, ‘\x06’, ‘\x16’, ‘\x00’, ‘\x11’, ‘\x00’, ‘\x00’, ‘K’, ‘G’, ‘\x00’, ‘\x0f’, ‘N’, ‘\x08’, ‘N’, ‘\x07’, ‘A’, ‘\x06’, ‘\x13’, ‘\x03’, ‘\x13’, ‘\x13’, ‘\x0b’, ‘\x01’, ‘\r’, ‘\x07’, ‘\t’, ‘\r’, ‘\x17’, ‘Y’, ‘L’, ‘\x0c’, ‘\x13’, ‘\t’]

Message 4: [‘\x1b’, ‘\x06’, ‘\x00’, ‘\x00’, ‘$’, ‘\x0f’, ‘\x17’, ‘\r’, ‘\x1b’, ‘\x00’, ‘\x00’, ‘T’, ‘C’, ‘\x08’, ‘\x1d’, ‘\x00’, ‘\x10’, ‘\x07’, ‘E’, ‘T’, ‘\x05’, ‘\n’, ‘S’, ‘\x03’, ‘O’, ‘\x1b’, ‘\x06’, ‘P’, ‘\x00’, ‘\x14’, ‘T’, ‘\x07’, ‘\x07’, ‘\x12’, ‘T’, ‘\n’, ‘I’, ‘\x01’, ‘\x15’, ‘\x08’, ‘\x0c’, ‘N’, ‘\x05’, ‘\x17’, ‘\x0f’, ‘E’, ‘\x01’, ‘\n’, ‘\x1e’, ‘M’, ‘+’, ‘\x01’, ‘T’, ‘M’, ‘\x08’, ‘\r’, ‘H’, ‘\x1d’, ‘\x06’, ‘\x00’, ‘\x00’, ‘\x0b’, ‘\x08’, ‘\x1d’, ‘T’, ‘\x0b’, ‘\x1d’, ‘Y’, ‘T’, ‘\x07’, ‘\x03’, ‘\x00’, ‘\x1f’, ‘\x1a’, ‘\x1f’, ‘\n’, ‘N’, ‘O’, ‘\x05’, ‘O’, ‘\x02’, ‘\x03’, ‘\x10’, ‘N’, ‘\x0c’, ‘\x1b’, ‘\x15’, ‘\x06’, ‘\x08’, ‘\x00’, ‘\x1c’, ‘D’, ‘\x1a’, ‘\x07’, ‘\x0f’, ‘\x11’, ‘\x1c’, ‘\x00’, ‘\x19’, ‘\t’, ‘\x0b’]

Message 5: [‘\x06’, ‘N’, ‘\x04’, ‘M’, ‘M’, ‘\x12’, ‘\x0c’, ‘\x1a’, ‘\x1b’, ‘\x17’, ‘E’, ‘T’, ‘\x0c’, ‘A’, ‘\x1d’, ‘A’, ‘\x1d’, ‘O’, ‘T’, ‘\x1c’, ‘\t’, ‘\x11’, ‘\x00’, ‘\x03’, ‘\x07’, ‘\x17’, ‘\x19’, ‘E’, ‘O’, ‘\x0f’, ‘S’, ‘F’, ‘\x1d’, ‘\t’, ‘\x1b’, ‘Y’, ‘M’, ‘\x1a’, ‘\x11’, ‘\x0c’, ‘I’, ‘\x1e’, ‘\x0e’, ‘\x1b’, ‘\x17’, ‘T’, ‘M’, ‘\x11’, ‘\x01’, ‘\x00’, ‘:’, ‘\x07’, ‘E’, ‘M’, ‘\x16’, ‘\n’, ‘\x1b’, ‘\x0c’, ‘\r’, ‘\x17’, ‘A’, ‘\x00’, ‘\n’, ‘N’, ‘T’, ‘\x0c’, ‘\x0e’, ‘T’, ‘T’, ‘\x04’, ‘\x0c’, ‘F’, ‘\x12’, ‘O’, ‘\x1b’, ‘\x18’, ‘\x00’, ‘\n’, ‘\x1e’, ‘T’, ‘\x06’, ‘\x00’, ‘\x06’, ‘T’, ‘E’, ‘\x17’, ‘\x1a’, ‘R’, ‘\x0e’, ‘\x1b’, ‘\x1a’, ‘\x01’, ‘\x1b’, ‘N’, ‘\x11’, ‘\x1e’, ‘\x18’, ‘N’, ‘\x08’, ‘\x15’, ‘\x1d’]

Message 6: [‘\x0c’, ‘\x06’, ‘\x04’, ‘M’, ‘\x1d’, ‘\x00’, ‘\x04’, ‘\x06’, ‘\x0c’, ‘N’, ‘\x0c’, ‘F’, ‘C’, ‘\x18’, ‘\x01’, ‘U’, ‘D’, ‘\x0e’, ‘R’, ‘\x11’, ‘H’, ‘\x16’, ‘E’, ‘\x12’, ‘\x04’, ‘\x1b’, ‘\x05’, ‘G’, ‘O’, ‘\x12’, ‘H’, ‘\x03’, ‘I’, ‘\x12’, ‘\x06’, ‘\x0c’, ‘T’, ‘\x07’, ‘R’, ‘\r’, ‘\x1a’, ‘N’, ‘\x03’, ‘\x17’, ‘\r’, ‘T’, ‘\x08’, ‘\x17’, ‘N’, ‘T’, ‘&’, ‘\x0e’, ‘N’, ‘M’, ‘\x00’, ‘C’, ‘\x04’, ‘\x00’, ‘\x0b’, ‘E’, ‘D’, ‘\x06’, ‘\x15’, ‘\x0b’, ‘C’, ‘\x10’, ‘\x00’, ‘R’, ‘T’, ‘!’, ‘\x11’, ‘\x00’, ‘\x12’, ‘\x01’, ‘\x11’, ‘\x04’, ‘U’, ‘\x1d’, ‘\x07’, ‘G’, ‘\n’, ‘\x1d’, ‘E’, ‘A’, ‘E’, ‘\x15’, ‘\x15’, ‘\x1c’, ‘A’, ‘\x1b’, ‘\x1d’, ‘D’, ‘\x0b’, ‘\x0b’, ‘A’, ‘\x17’, ‘\x01’, ‘P’, ‘\x0c’, ‘\x0f’, ‘\x1d’]

Message 7: [‘\r’, ‘\x1b’, ‘\x0c’, ‘L’, ‘\t’, ‘\x08’, ‘\r’, ‘\x0f’, ‘I’, ‘\x1a’, ‘\x00’, ‘C’, ‘\x0b’, ‘\x0f’, ‘\x07’, ‘C’, ‘\x05’, ‘\x03’, ‘\x00’, ‘\x07’, ‘\x11’, ‘\x16’, ‘T’, ‘\x12’, ‘\x02’, ‘\x01’, ‘K’, ‘I’, ‘\x01’, ‘\x10’, ‘O’, ‘\n’, ‘\x1f’, ‘\x03’, ‘\x07’, ‘Y’, ‘A’, ‘O’, ‘\x1e’, ‘\x0b’, ‘\x1d’, ‘N’, ‘\x0e’, ‘\x14’, ‘Y’, ‘H’, ‘\x0c’, ‘\x17’, ‘\n’, ‘\x00’, ‘9’, ‘\x00’, ‘R’, ‘\x06’, ‘A’, ‘\x02’, ‘\x06’, ‘\r’, ‘N’, ‘\x16’, ‘P’, ‘\x06’, ‘\x02’, ‘\x07’, ‘A’, ‘\x08’, ‘\x06’, ‘Z’, ‘\x11’, ‘\x0c’, ‘E’, ‘K’, ‘\x19’, ‘\x00’, ‘\x05’, ‘\x07’, ‘E’, ‘\x0b’, ‘\x01’, ‘E’, ‘O’, ‘\x02’, ‘\x04’, ‘N’, ‘\x02’, ‘\r’, ‘\x15’, ‘\x15’, ‘\x04’, ‘\x1c’, ‘R’, ‘\x05’, ‘\x07’, ‘\n’, ‘A’, ‘\x02’, ‘\x0b’, ‘O’, ‘\x19’, ‘\x0e’, ‘\r’]

Message 8: [‘\x06’, ‘\x1a’, ‘\x16’, ‘\x00’, ‘\x04’, ‘\x0c’, ‘\x13’, ‘\x07’, ‘\x1a’, ‘\x1d’, ‘\x0c’, ‘B’, ‘\x0f’, ‘\x04’, ‘N’, ‘T’, ‘\x0b’, ‘O’, ‘M’, ‘\x1b’, ‘\x1e’, ‘\x00’, ‘\x00’, ‘\x03’, ‘\x00’, ‘R’, ‘\x07’, ‘I’, ‘\x19’, ‘\x03’, ‘\x00’, ‘\x12’, ‘\x06’, ‘F’, ‘\x1b’, ‘\t’, ‘E’, ‘\x1d’, ‘\x13’, ‘\x10’, ‘\x0c’, ‘N’, ‘\x00’, ‘\x06’, ‘Y’, ‘A’, ‘\x03’, ‘\x1c’, ‘N’, ‘L’, ‘+’, ‘\x19’, ‘E’, ‘\x01’, ‘A’, ‘\x14’, ‘\x01’, ‘\x1d’, ‘\x06’, ‘\n’, ‘U’, ‘\x17’, ‘A’, ‘\x02’, ‘E’, ‘\x05’, ‘\x19’, ‘I’, ‘\x1a’, ‘\x0f’, ‘E’, ‘T’, ‘\x05’, ‘\x0e’, ‘\x11’, ‘\x0e’, ‘S’, ‘O’, ‘\x04’, ‘I’, ‘\x1b’, ‘\x1d’, ‘E’, ‘S’, ‘\x00’, ‘\x1d’, ‘\x19’, ‘\x1b’, ‘\x0f’, ‘\x08’, ‘\x1e’, ‘\x1d’, ‘I’, ‘\x03’, ‘\x04’, ‘\x13’, ‘\x17’, ‘I’, ‘\x03’, ‘\x06’, ‘\x02’]

Message 9: [‘\x1b’, ‘\x06’, ‘\x00’, ‘\x00’, ‘:’, ‘\x04’, ‘\x01’, ‘H’, ‘\x08’, ‘\x1d’, ‘E’, ‘i’, ‘C’, ‘\x04’, ‘\x00’, ‘V’, ‘\r’, ‘\x1c’, ‘A’, ‘\x13’, ‘\r’, ‘\x01’, ‘\x00’, ‘\x1e’, ‘\x1b’, ‘R’, ‘\x1c’, ‘E’, ‘O’, ‘\x0e’, ‘A’, ‘\x10’, ‘\x0c’, ‘F’, ‘\x1a’, ‘\x16’, ‘T’, ‘O’, ‘\x01’, ‘\x01’, ‘\x0c’, ‘\x00’, ‘A’, ‘\x1b’, ‘\r’, ‘\x00’, ‘\x14’, ‘\x00’, ‘\x1a’, ‘\x00’, ‘\x1a’, ‘\x07’, ‘E’, ‘M’, ‘\x07’, ‘\x16’, ‘\x1c’, ‘\x1c’, ‘\x1c’, ‘\x00’, ‘\x00’, ‘\n’, ‘\x12’, ‘N’, ‘S’, ‘\x10’, ‘\x06’, ‘L’, ‘\x18’, ‘H’, ‘\x16’, ‘O’, ‘W’, ‘\x02’, ‘\x07’, ‘\x08’, ‘H’, ‘O’, ‘\x04’, ‘I’, ‘\x08’, ‘\t’, ‘\x00’, ‘R’, ‘E’, ‘\x0c’, ‘\x1c’, ‘\x13’, ‘\x0f’, ‘O’, ‘\x06’, ‘\x0c’, ‘\x0c’, ‘N’, ‘\x11’, ‘\x13’, ‘\n’, ‘T’, ‘M’, ‘5’, ‘\x07’]

Message 10: [‘\x06’, ‘N’, ‘\x10’, ‘S’, ‘\x08’, ‘\x05’, ‘C’, ‘\x1c’, ‘\x06’, ‘N’, ‘\x11’, ‘H’, ‘\n’, ‘\x0f’, ‘\x05’, ‘\x00’, ‘\x10’, ‘\x07’, ‘A’, ‘\x00’, ‘H’, ‘\x06’, ‘Y’, ‘\x15’, ‘\n’, ‘\x00’, ‘\x18’, ‘P’, ‘\x0e’, ‘\x05’, ‘E’, ‘F’, ‘\x1e’, ‘\x07’, ‘\x07’, ‘Y’, ‘F’, ‘\x06’, ‘\x14’, ‘\x10’, ‘\x10’, ‘N’, ‘\x18’, ‘\x17’, ‘\x18’, ‘R’, ‘\x1e’, ‘E’, ‘\x0f’, ‘W’, ‘/’, ‘\x16’, ‘\x00’, ‘:’, ‘\t’, ‘\x02’, ‘\x1c’, ‘I’, “‘“, ‘E’, ‘T’, ‘\x0b’, ‘\x0e’, ‘\x1b’, ‘G’, ‘\x0c’, ‘\x1b’, ‘\x00’, ‘\x03’, ‘\t’, ‘\x16’, ‘\x00’, ‘\x11’, ‘\x06’, ‘\x14’, ‘\x1f’, ‘Y’, ‘O’, ‘\x1f’, ‘E’, ‘\x0e’, ‘\x1c’, ‘\x16’, ‘\x00’, ‘\x04’, ‘\x0f’, ‘\x15’, ‘\x0b’, ‘A’, ‘\x18’, ‘\x13’, ‘\x17’, ‘I’, ‘\x01’, ‘\x0f’, ‘\x1e’, ‘\x00’, ‘\x00’, ‘\x19’, ‘\x04’, ‘\x00’]

Message 11: [‘\x18’, ‘\x0b’, ‘E’, ‘A’, ‘\x1f’, ‘\x04’, ‘C’, ‘\x1c’, ‘\x01’, ‘\x0b’, ‘E’, ‘C’, ‘\x0b’, ‘\x08’, ‘\x02’, ‘D’, ‘\x16’, ‘\n’, ‘N’, ‘T’, ‘\x07’, ‘\x03’, ‘\x00’, ‘\x16’, ‘O’, ‘\x06’, ‘\x0e’, ‘C’, ‘\x07’, ‘\x08’, ‘O’, ‘\n’, ‘\x06’, ‘\x01’, ‘\x1d’, ‘\x1a’, ‘A’, ‘\x03’, ‘R’, ‘\x05’, ‘\x0e’, ‘\x0b’, ‘A’, ‘%’, ‘\x1c’, ‘\x00’, ‘\x05’, ‘\x04’, ‘\x18’, ‘E’, ‘n’, ‘\t’, ‘O’, ‘\x18’, ‘\x0f’, ‘\x07’, ‘H’, ‘\x1a’, ‘\x1a’, ‘\x17’, ‘E’, ‘\x02’, ‘\x0c’, ‘\x02’, ‘I’, ‘\n’, ‘\n’, ‘D’, ‘T’, ‘\x1f’, ‘\x04’, ‘Y’, ‘\x04’, ‘O’, ‘\x1d’, ‘\r’, ‘\x00’, ‘\x0b’, ‘\t’, ‘I’, ‘\x01’, ‘\t’, ‘E’, ‘M’, ‘\x10’, ‘\x1b’, ‘\x1c’, ‘R’, ‘\x0e’, ‘\t’, ‘R’, ‘\x0b’, ‘\x1c’, ‘\x1c’, ‘A’, ‘\x00’, ‘\x16’, ‘U’, ‘\x19’, ‘\x08’, ‘\x00’]

Message 12: [‘\x1c’, ‘\x1a’, ‘\x1c’, ‘L’, ‘\x08’, ‘A’, ‘\x16’, ‘\x1b’, ‘\x0c’, ‘\n’, ‘E’, ‘T’, ‘\x0c’, ‘A’, ‘\x0c’, ‘E’, ‘D’, ‘\x0e’, ‘N’, ‘T’, ‘\x01’, ‘\x0b’, ‘T’, ‘\x12’, ‘\x1d’, ‘\x13’, ‘\x08’, ‘T’, ‘\x06’, ‘\t’, ‘N’, ‘F’, ‘\x0b’, ‘\x03’, ‘\x00’, ‘\x0e’, ‘E’, ‘\n’, ‘\x1c’, ‘D’, ‘\x1d’, ‘\x06’, ‘\x04’, ‘R’, ‘\x11’, ‘U’, ‘\x00’, ‘\x04’, ‘\x00’, ‘\x00’, ‘=‘, ‘\x00’, ‘U’, ‘\x01’, ‘A’, ‘\x02’, ‘\x06’, ‘\r’, ‘N’, ‘\x11’, ‘O’, ‘\x0c’, ‘\r’, ‘\x1d’, ‘\x00’, ‘\x10’, ‘\x07’, ‘A’, ‘\x00’, ‘H’, ‘\x12’, ‘E’, ‘\x05’, ‘\n’, ‘R’, ‘\x07’, ‘I’, ‘\x02’, ‘\x0f’, ‘T’, ‘\x06’, ‘\x00’, ‘\x02’, ‘\x00’, ‘,’, ‘\x16’, ‘T’, ‘\x06’, ‘\t’, ‘\n’, ‘R’, ‘\x00’, ‘\x00’, ‘\t’, ‘\x08’, ‘\x06’, ‘\x18’, ‘L’, ‘M’, ‘\x04’, ‘\x1c’]

The Key/Secret Message is as follows:

[‘Space’, ‘n’, ‘e’, ‘Space’, ‘m’, ‘a’, ‘c’, ‘h’, ‘i’, ‘n’, ‘e’, ‘Space’, ‘c’, ‘a’, ‘n’, ‘Space’, ‘d’, ‘o’, ‘Space’, ‘t’, ‘h’, ‘Space’, ‘Space’, ‘w’, ‘o’, ‘r’, ‘k’, ‘Space’, ‘o’, ‘f’, ‘Space’, ‘f’, ‘i’, ‘f’, ‘t’, ‘y’, ‘Space’, ‘o’, ‘r’, ‘d’, ‘i’, ‘n’, ‘a’, ‘r’, ‘y’, ‘Space’, ‘m’, ‘e’, ‘n’, ‘Space’, ‘n’, ‘Space’, ‘Space’, ‘m’, ‘a’, ‘c’, ‘h’, ‘i’, ‘n’, ‘e’, ‘Space’, ‘Space’, ‘a’, ‘n’, ‘Space’, ‘d’, ‘Space’, ‘Space’, ‘t’, ‘h’, ‘e’, ‘Space’, ‘w’, ‘o’, ‘r’, ‘k’, ‘Space’, ‘o’, ‘f’, ‘Space’, ‘o’, ‘n’, ‘e’, ‘Space’, ‘e’, ‘Space’, ‘t’, ‘r’, ‘a’, ‘o’, ‘r’, ‘d’, ‘i’, ‘n’, ‘a’, ‘Space’, ‘y’, ‘Space’, ‘m’, ‘a’, ‘Space’]

The Key/Secret Message Vertically:

0 Space

1 n

2 e

3 Space

4 m

5 a

6 c

7 h

8 i

9 n

10 e

11 Space

12 c

13 a

14 n

15 Space

16 d

17 o

18 Space

19 t

20 h

21 Space

22 Space

23 w

24 o

25 r

26 k

27 Space

28 o

29 f

30 Space

31 f

32 i

33 f

34 t

35 y

36 Space

37 o

38 r

39 d

40 i

41 n

42 a

43 r

44 y

45 Space

46 m

47 e

48 n

49 Space

50 n

51 Space

52 Space

53 m

54 a

55 c

56 h

57 i

58 n

59 e

60 Space

61 Space

62 a

63 n

64 Space

65 d

66 Space

67 Space

68 t

69 h

70 e

71 Space

72 w

73 o

74 r

75 k

76 Space

77 o

78 f

79 Space

80 o

81 n

82 e

83 Space

84 e

85 Space

86 t

87 r

88 a

89 o

90 r

91 d

92 i

93 n

94 a

95 Space

96 y

97 Space

98 m

99 a

100 Space

The secret Message is as follows:

One machine can do the work of fifty ordinary men; no machine can do the work of one extraordinary man