**CS411-507 2021-2022 FALL HOMEWORK 3**

**Elif Cemre Durgut – 26493**

In addition to the folder, you can find my codes in the following Colab link as well:

<https://colab.research.google.com/drive/1zOMHGuMnpqSCWVk2FmLSBzT2lvtHT5OH?usp=sharing>

**Question 1)**

The formula is C = M^e (mod N). According to hint, if M << N, M^e is also smaller than N. So, we do not need to check the modulus while finding M. First, I found a lower limit. To find it, I calculated the number of digits in C (1147) and then I tried some numbers (say m) to find a kind of upper limit as m^e such that #of digits is 1146 so that it does not exceed the C. This number is 2x10^69. Then, I developed an algorithm such that it starts from the second digit and changes that digit to 9, if the updated number^e exceeds C, then it lowers 9 to 8, and does the same operation. Else, it saves the number with the update and moves on to the next digit. To sum up, this algorithm finds each digit separately by comparing the number with C. In the end, I double-checked by substituting and it is correct.

M = 26959946667150639794667015087019630673637144422540572481103610249216

**Question 3)**

I used the LFSR function to find the output of connective polynomials then I compared these bits with the given z. I did this operation with different initial states(there are 2^L initial states, L: highest degree of the polynomial) and I stored the max number of matches and the corresponding initial state.

1. Maximum number of matches: 90 out of 90

Initial state: 0111110

1. Maximum number of matches: 62 out of 90

Initial state: 00000001000

1. Maximum number of matches: 64 out of 90

Initial state: 0001110100110

**Question 5)**

I used BitVector module of Python. First, I created BitVector object, then I used gf\_multiply\_modular function for part a and gf\_MI function for part b by looking at how the module functions are used from <https://engineering.purdue.edu/kak/dist/BitVector-3.5.0.html>.

**Question 6)**

Firstly, I tried to reduce the digests and compared it with P(j, t-1) values which are the second column of rainbow.txt. And I could not find a match, so I applied f function (hash + reduce) to the first column of rainbow.txt file and I compared the digests. I repeat the same until I find a match. I wrote the code, but I could not find the answers ☹