Mark Shinozaki
Cpts 427 – Skittles

1. Introduction

- Overview of the puzzle: The puzzle that was included in the assignment, also called Skittles, is a file that contains a list that contains 4167 Roman Numerals and 17789 characters in total. This sort of puzzle is commonly found in CTF (Capture the flag) games or puzzles, where each numeral could represent a number, letter, sequence of bits which all depends on the nature of the encryption or encoding method used.

- Objective: The goal is ultimately to solve the puzzle, but in this case and the scope of this assignment Ill try my best to provide an analysis and breakdown to potentially decode the message or find a hidden flag.

2. Methodology
    - Initial Analysis:
        - My first impression is obviously that the entire puzzle is a sequence of roman numerals. Also, there is no direct pattern that I can detect without reviewing more exact data or statistics using an algorithm. It does appear that some of the Roman numerals appear more frequently than others and that some appear maybe less than 10 times compared to other Roman Numerals which appear more than 20-50 times throughout the puzzle.
        - Upon a more extensive review I found that there is possibly a delimiter or basic text key that is used for sorting which would be "M" which for this puzzle could represent a flag or something to pay direct attention but also stands for 1000 in the typical roman numeral system. Also, I've noticed that each Roman Numeral translates to a value between 1 (I) and 1000 which is (M).
        - The ideas here being a few core crypography concepts including, Value-Based Encoding, meaning, each numeral could represent an ASCII value, or a shift in ASCII values, translating numerals directly to characters based on their numeric values. Positional Encoding, The position of each numeral or group of numerals separated by "M" might represent different characters or commands in a more complex encoding scheme. Theres also, Symbolic representation, "M" might not only act as a delimiter but could also modify the meaning of the numerals around it, prerhaps indicting a different base in number systems or a shift in encoding style.
    - Tools and Resources Used:
        - To try and tackle or figure out this complex puzzle, I will be using the Skittles document and VScode to try three possibilities. Statistical Analysis, which would be Calculating the frequency of each numeral and "M" to understand their distribution
3. Problem Solving process
    - Conversion of Data
        - High frequency may indicate common letters or separators. Segmentation, splitting the sequence at each "M" to see if the segments can be decoded individually, possibly revealing patterns or coherent segments of data.

Mark Shinozaki
Cpts 427 – Skittles

Numerical Conversion, Convert the Roman numerals to their corresponding numbers and apply various cryptographic encoding to try and decode the puzzle, which includes, ASCII conversion, or Shift ciphers.

```
-   def roman_to_int(s):
-       roman_dict = {'I': 1, 'V': 5, 'X': 10, 'L': 50, 'C': 100, 'D': 500,
    'M': 1000}
-       total = 0
-       prev_value = 0
-       for char in reversed(s.upper()):
-           current = roman_dict[char]
-           if current >= prev_value:
-               total += current
-           else:
-               total -= current
-           prev_value = current
-       return total
-
-   def read_roman_numerals(file_path):
-       with open(file_path, 'r') as file:
-           content = file.read().replace('\n', ' ')
-       return content
-
-   def split_numerals(content):
-       segments = content.split(' M ')
-       return segments
-
-   def convert_numerals_to_numbers(segments):
-       numbers_list = []
-       for segment in segments:
-           numerals = segment.split()
-           numbers = [roman_to_int(numeral) for numeral in numerals]
-           numbers_list.append(numbers)
-       return numbers_list
-
-   def calculate_frequency(content):
-       numerals = content.split()
-       frequency = {}
-       for numeral in numerals:
-           if numeral in frequency:
-               frequency[numeral] += 1
-           else:
-               frequency[numeral] = 1
-       return frequency
-
-   file_path = 'RomanNum.txt'
```

Mark Shinozaki
Cpts 427 – Skittles

```
-   roman_content = read_roman_numerals(file_path)
-   frequency = calculate_frequency(roman_content)
-   segments = split_numerals(roman_content)
-   numbers_list = convert_numerals_to_numbers(segments)
-
-   print("Frequency of Roman Numerals:", frequency)
-   print("Converted Numbers:", numbers_list)
-
```

- Decoding Attempts
    - (Attempt 1)
    - I ran the code above to get a better idea of the puzzle and gain some insight into cracking it, here is one of the outputs that I got from the Frequency of Roman Numerals: {'CXVII': 24, 'XXVII': 121, 'XIV': 102, 'VI': 195, 'XVII': 89, 'CV': 108, 'XXIII': 134, 'CXIV': 107, 'XIII': 98, 'XXIV': 83, 'XI': 103, 'XV': 71, 'XVI': 103, 'CXVIII': 112, 'I': 113, 'XXVI': 89, 'XX': 60, 'X': 137, 'VII': 117, 'XXV': 45, 'XXVIII': 106, 'IX': 43, 'CXXVI': 24, 'CXI': 106, 'VIII': 87, 'XXII': 105, 'V': 82, 'XII': 127, 'XXI': 73, 'CIII': 138, 'XIX': 113, 'M': 119, 'LXXVII': 15, 'XXXI': 51, 'CXXI': 112, 'CXXVII': 2, 'III': 98, 'II': 112, 'XXIX': 143, 'XCVIII': 91, 'CI': 25, 'XXX': 77, 'CIX': 4, 'LXXXVIII': 13, 'IV': 76, 'XVIII': 36, 'CXX': 6, 'LXIX': 8, 'CXXV': 3, 'CX': 20, 'CVIII': 10, 'XCII': 15, 'CIV': 9, 'CXIII': 8, 'CXXII': 14, 'XCVI': 8, 'CVI': 1, 'LXXII': 8, 'CXXIV': 6, 'XCVII': 5, 'LXXXIII': 12, 'XCIX': 13, 'LXVII': 14, 'CXIX': 3, 'CXII': 1, 'CXVI': 5, 'C': 2, 'CVII': 4, 'XLVII': 1, 'CXXIII': 1, 'CII': 1}. Then separately, I got a list of all of the Converted Numbers, here is an example of the values and output that I received, Converted Numbers: [[117, 27, 14, 6, 17, 105, 23, 114, 13, 24, 11, 15, 16, 118, 1, 26, 20, 10, 7, 27, 118, 23, 25, 28, 9, 11, 7, 17, 126, 111, 24, 9, 6, 105, 23, 114, 8, 22, 8, 6, 5, 15, 114, 12, 11, 8, 21, 13, 118, 5, 14, 10, 103, 22, 1, 19], [10, 117, 77, 21, 1, 31, 17, 7, 121, 16, 7, 5, 26, 127, 1, 16]. In terms of what this tells me, is there a mix of values and numbers but since this is the first attempt I'm not gaining a very significant understanding of the puzzle quite yet until I do further analysis.
    - (Attempt 2)
    - In this next attempt, Ill be using the values I received to get an ASCII output, which can better help me understand the puzzle. uirvv~oirrvg
        - uMy
        - ii~b
        - evyv
        - vyvmXbrvg
        - ygoyvbuMi
        - rvvyv
        - i
        - grxXE~oiov}nv
        - rgvubyvioln\uigv}nvrxrhv~oi

        - ovqXhzrevge~oby`ibrb
        - yjH
        - v
        - qorbyb
        - ugqoobvln\Xirbyyri
    - This is a sample of the data that I received as the output, it is random gibbersh and still doesn't directly give us solution to the puzzle but it does give me some insight and I can take this approach one step further in the next step to see if that yields any better understanding,
    - (Attempt 3)
    - I have now converted the values to ascii and then analyzed the frequency of the data or the values that I received in the previous attempt and now I have implemented a Caesar Cipher and Vigenre Cipher to try and get a better understanding of the puzzle and this gives me better of different keys and shifts systematically. After Decrpying with different shifts and key words, Im still a stuck and not exactly sure what the solution to the puzzle is.
    - (Attempt 4)
    - I tried using Skittles as a keyword in the Vigenère Cipher and received this data back,

```
c?JCCSK7`HN4B"U7WT:2CNRDgLE%/GR5g=GFC7QHNCFKE5`N0↕KDE=dSFC?<R
CPON6<AJ*c??CJCR@fH@CKDE=dG05GGADU;V</N<7PH:F7↔R?]H:F/JC?]E:C9C8
&WH:FFGEDPR ?2KUGWT?C?5E7gO9 -8UL]=GBFNU=PO:4<↑K0`??/4DC5e#
;4KK@POMF?NN0g;N4C:R5Q)JFC7NDUEN/<GC*k?Q</G>7U=Q?FB4G`8?C;>KH`=
J/?A▼*cPJK<FK=POAF<>UDj▲Q?FGE5UDGC?<80]=HF2KK<`=LF6K>=[.?F6ON5N?
?K<>K@P=9 →:MG`8N6;>U0c=AG?A▼FUE:6GG>?gHAB4K!D]??66NR5d=QA
KK3d8Q64NX=SL:?<Q!!NEA0FONG`=J64D>5]↓?44KK7UL?CF↑C0RL=C2KC5]?Q4;>
SOMFKDV@PPJ4<NEE;▲FK6KUGgL@4<7UD;??CFKEGd=??9↑K5];N/C<ECUE;F0NR
HF8G6?DN5]LA4)-
D5`O?<6NR0PHJ/B"N5Y8G6<:R5gOT↕4<K0`TG6F<K0c#J46NR0`?A<E-
!LPO:6C5R<WG?4?♦N5OT%C?<>7WOG/?:85]=JF<IC@d=Q?/(N7ULQ0FK>GWB7D5
`?J567UD]?+<C7R3d=??6NN*U;Q2C<NG]8;FCK>0Y▲GC4<KGlE:6<7H◀JK=C4GE
GPON/FGAD\↓?64DE5c=J4CN▼5]LA4?DW<WO:/?NI!!PTGK?CEG]H:64KH◀JKQC?
<EL]?G/0(E@R?M46>KH`8?4?A▼GP?;FC7N5S2:54D>Gd=G64<=!;;J;?>CDP9QCC<
UD1EG6/G?GdEA4/"N0d8J?/7CD\↓Q<4GCDUL??4146_EA?6DZ=fHF6?KC@h.Q4BN
E3d=AFC<89PT:<0NB5l8N<F>UHF;N64MN6W=QC6<CJ3)8K<ON5`?GC?D<DgT↔6
B<K@WHG/<K>5d?
F6<U7POG64GU=b#JC4GNG`8A9;14=lEA4F>G0]=+66GE5]H:<CNR76K?CFNC=d=:
B→DC0\?G<6<C@]J?. "KGS
```

    - Challenges Encountered
        - Complexity and Ambiguity, the puzzle inherent complexity and lack of clear direction might have been the biggest challenges. The use of Roman numerals and their conversion into numbers that don't immediately suggest a pattern or decipherable code adds layers of ambiguity.

        - Decoding Methods, when deciding which cryptographic decoding method to use, such as ASCII conversion, Caesar Cipher or Vigenere Cipher, and not obtaining clear and meaningful results from them is definitely frustrating, the results often seemed random and didn't provide significant insight and lead to questioning the methods and assumptions about the puzzle.

4. Insights and Learning
    - Observations
        - High frequency and Distribution Analysis, One of the key findings from my intital statistical analysis was the high frequency of certain Roman Numerals over others. This could hint at common letters or symbols in a coded message. The Delimiter Role of 'M', Identifying M as a potential delimieter or a key symbol was an important observation which suggested that the numerals might be segmented or interpreted differently before and after each M. The ASCII outputs, the ASCII conversion resulted in gibberish or non-meaningful outputs, it showed that simple direct conversion might not be the solution, hinting at more complex encoding schemes or the need for additional decoding layers.
        - Reflection and conclusion, This puzzle serves as a robust example of how complex problem-solving in cryptanalysis can often be more about the journey and the process than about reaching the conclusion or solving the puzzle. Using persistent methods to tackle the puzzle helped me in understanding the muti-layered encryption scenarios or more obscure ciphers. Even though I didn't figure out the cipher, the groundwork is there to modify or alter my approach to try a more practical approach to solving the puzzle.