IceCTF 2016 – Over the Hill

Problem

> Over the hills and far away... many times I've gazed, many times been bitten. Many dreams come true and some have silver linings, I live for my dream of a decrypted flag.

Solution

As hinted at by the name, this challenge uses the Hill Cipher.

Enter the alphabet, matrix and ciphertext values provided with the problem.

```
alphabet = "abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ123456789_{}";

matrix = {
  {54, 53, 28, 20, 54, 15, 12, 7},
  {32, 14, 24, 5, 63, 12, 50, 52},
  {63, 59, 40, 18, 55, 33, 17, 3},
  {63, 34, 5, 4, 56, 10, 53, 16},
  {35, 43, 45, 53, 12, 42, 35, 37},
  {20, 59, 42, 10, 46, 56, 12, 61},
  {26, 39, 27, 59, 44, 54, 23, 56},
  {32, 31, 56, 47, 31, 2, 29, 41}};

ciphertext = "7Nv7}dI9hD9qGmP}CR_5wJDdkj4CKxd45rko1cj51DpHPnNDb__EXDotSRCP8ZCQ";
```

Get the index of each ciphertext character using the alphabet. We assume 0 based indexes (Mathematica uses 1 based indexes normally. Wikipedia used 0 based indexes and the highest index in the key was 63, so this challenge probably uses 0 based indexes.

`Characters[]` will split a string into a list of single character strings, whose index can be looked up in alphabet using `StringPosition[]`.

```
ciphertextIndexes = StringPosition[alphabet, #][[1, 1]] - 1 & /@ Characters@ciphertext
```

```
{58, 39, 21, 58, 63, 3, 34, 60, 7, 29, 60, 16, 32, 12, 41, 63, 28,
43, 61, 56, 22, 35, 29, 3, 10, 9, 55, 28, 36, 23, 3, 55, 56, 17, 10,
14, 52, 2, 9, 56, 52, 29, 15, 33, 41, 13, 39, 29, 1, 61, 61, 30, 49,
29, 14, 19, 44, 43, 28, 41, 59, 51, 28, 42}
```

Split the ciphertext into vertical matrices the size of the key matrix.

```
ciphertextParts = Partition[List /@ ciphertextIndexes, Length@matrix]
```

```
{{{58}, {39}, {21}, {58}, {63}, {3}, {34}, {60}}, {{7}, {29}, {60},
{16}, {32}, {12}, {41}, {63}}, {{28}, {43}, {61}, {56}, {22}, {35},
{29}, {3}}, {{10}, {9}, {55}, {28}, {36}, {23}, {3}, {55}}, {{56},
{17}, {10}, {14}, {52}, {2}, {9}, {56}}, {{52}, {29}, {15}, {33},
{41}, {13}, {39}, {29}}, {{1}, {61}, {61}, {30}, {49}, {29}, {14},
{19}}, {{44}, {43}, {28}, {41}, {59}, {51}, {28}, {42}}}
```

Just do a dot product of the message parts with the inverse of the matrix. Everything is done with a modulus of the length of the alphabet.

```
messageParts = Mod[
  Inverse[matrix, Modulus -> StringLength@alphabet].#,
  StringLength@alphabet] & /@ ciphertextParts
```

```
{{{34}, {2}, {4}, {28}, {45}, {31}, {62}, {11}}, {{8}, {13}, {4},
{0}, {17}, {61}, {0}, {11}}, {{6}, {4}, {1}, {17}, {0}, {61}, {15},
{11}}, {{20}, {18}, {61}, {11}, {4}, {3}, {61}, {25}}, {{4}, {15},
{15}, {4}, {11}, {8}, {13}, {61}}, {{0}, {17}, {4}, {61}, {0}, {61},
{1}, {4}}, {{0}, {20}, {19}, {8}, {5}, {20}, {11}, {61}}, {{12},
{52}, {23}, {19}, {20}, {17}, {4}, {63}}}
```

`Flatten[]` the nested list of indexes into a one dimensional list, then convert the indexes back into characters, and join the single character strings back together.

```
StringJoin[StringPart[alphabet, # + 1] & /@ Flatten@messageParts]
```

```
IceCTF{linear_algebra_plus_led_zeppelin_are_a_beautiful_m1xture}
```