British National Cipher Challenge 2019
Challenge 9B (final challenge)

solution by madness

The ciphertext for this challenge is a long string of 0s, 1s, and 2s. It is too long to include here, but it starts off like this:

```
11101 00101 00001 21122 10001 10111 00101 10001 01111 00210 11011
11111 01011 01001 01011 21001 11000 01011 01000 00012 21222 10011
11010 01000 01001 10201 00000 11101 10010 01011 00012 00212 00011
01000 01101 01110 00000 20021 10010 00100 11010 11011 00002 10000
10000 11010 00001 00022 00110 00101 10101 10001 01100 00210 01111
10010 01001 00000 10111 02020 00101 00101 01001 11000 01022 21210
00001 00110 10101 00010 21020 02111 00100 ...
```

The first thing to do is a bit of numerology on the number of digits. There are 18,480 digits, and this number has a prime factorization of $2 \times 2 \times 2 \times 2 \times 3 \times 5 \times 7 \times 11$. From challenge 9A (plaintext in the Appendix), we know that the keyword NIOBE was used. In Greek mythology, Niobe grieved the deaths of her seven sons and seven daughters. Aha! Seven is also one of the factors of the ciphertext length.

Now to look for patterns in the ciphertext. I work on a linux terminal, so some of the command I use may seem unfamiliar. But what this series of commands does is print the ciphertext without spaces and colors all of the 2s red.

```
cat ciphertext.txt | tr -d ' ' | grep --color 2
```

You can see that a pattern emerges, as the 2s form bands across the text:

```
11101001010000121122100011011100101100010111100210110111111101011010010101121001110000 10
11010000001221222100111101001000010011020100000111011001001011000120021200011010000110 10
11100000020021100100010011010110110000210000100000110100000100022001100010110101100010110
00021001111100100100100000101110202000101001010100111000010222121000001001101010100010210
02002111001000011011101100002101201110110011001100000100211001101101011000001100002100 11
11100100100100100101100200002100001010100101001001100022101100011101111000000120121020 10
01100011001010110010210201101111001010000010001100222000000100010100110001110210210010 10
01001110110000210021211111010001100010111000201202010101100100010011110001100101000101111
01000000012011220011100001111101000000021122100100110100101001001022002000011000001101 11
10100100210010100011110100000000101202202000001110110010000100222100211011100000100011 10
00200120110000011010100100011110221001000101011110100010010221220100000011010 ...
```

The bands come roughly with a period of 28, so next I looked at each 28-digit block, and again highlighted the 2s.

```
cat ciphertext.txt | tr -d ' ' | \
grep -o ............................ | grep --color 2
```

What I see here is that the 2s only occur in the third sub-block of seven digits in each block of 28.

```
1110100101000012112210001101
1100101100010111100210110111
```

```
11110101101001010112100111100
00101101000000122122210011111
01001000010011020100000111101
10010010110001200212000110100
00011010111000000200211001000
01001101011011000021000010000
01101000001000220011000101100
10110001011000021001111100100
01001000001011102020001010010
01010011100001022212100000100
01101010100010210200211100100
00011011101100002101201110110
00110011000001002110011011010
01100000110000210011111001000
10010010010110020000210000100
...
```

I had been thinking about binary and ternary (base-3) encoding for the last four hours, so at this point it came to me in a flash: The text was encoded in a hybrid number system in which the first, second, and fourth digit of each number is binary, while the third digit is ternary. This would allow the encoding of $2 \times 2 \times 3 \times 2 = 24$ letters. That should be enough for most texts.

The most reasonable way to unpack a 28-digit block of digits into seven four-digit numbers, when they have been shuffled like the ciphertext, is to divide the block into four seven-digit sub-blocks and take the first digit of each block, then the second digit of each block, *ad finem*. For example, the first block is

```
1110100 1010000 1211221 0001101
```

Taking the first digit of each sub-block gives 1110. The second digit of each block gives 1020. The remaining five numbers are 1110, 0011, 1021, 0020, and the lonely 0011, who doesn't get a color.

Next, I evaluated each number in decimal (base 10) and assigned a letter to each. To convert a number in this hybrid system to decimal, follow this formula:

$$X = 12 \times (\text{first digit}) + 6 \times (\text{second digit}) + 2 \times (\text{third digit}) + (\text{fourth digit})$$

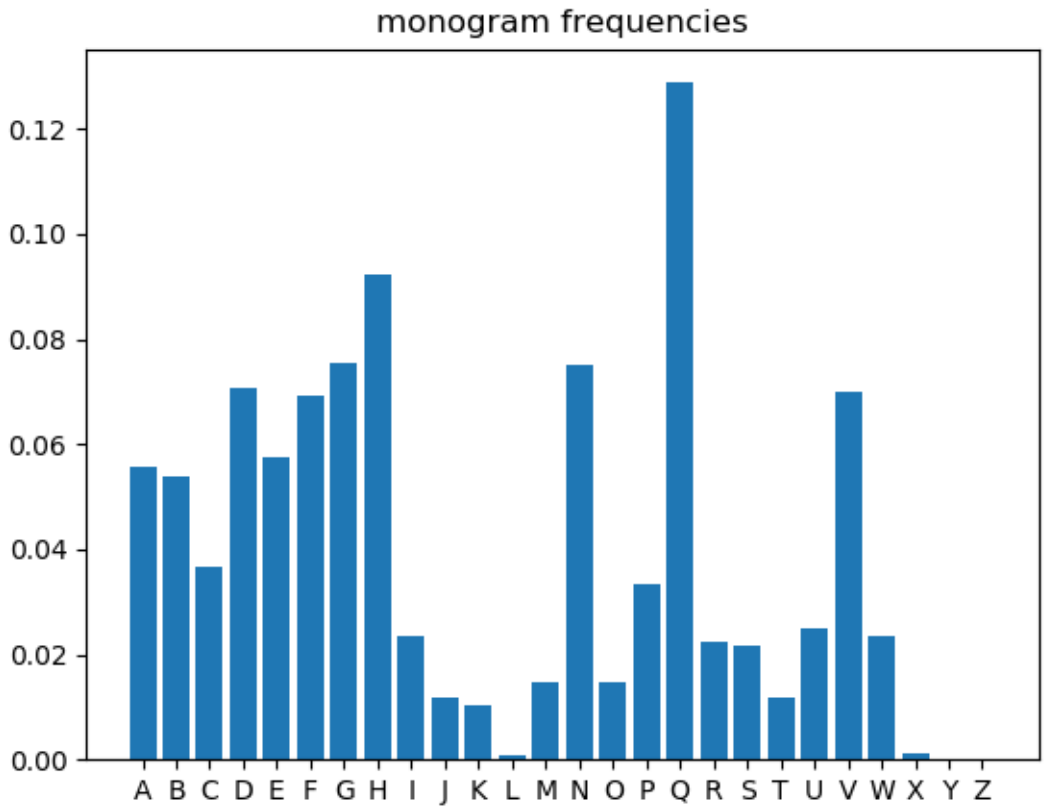
So my assignment of letters is as follows:

0000 = 0 → A	0110 = 8 → I	1020 = 16 → Q
0001 = 1 → B	0111 = 9 → J	1021 = 17 → R
0010 = 2 → C	0120 = 10 → K	1100 = 18 → S
0011 = 3 → D	0121 = 11 → L	1101 = 19 → T
0020 = 4 → E	1000 = 12 → M	1110 = 20 → U
0021 = 5 → F	1001 = 13 → N	1111 = 21 → V
0100 = 6 → G	1010 = 14 → O	1120 = 22 → W
0101 = 7 → H	1011 = 15 → P	1121 = 23 → X

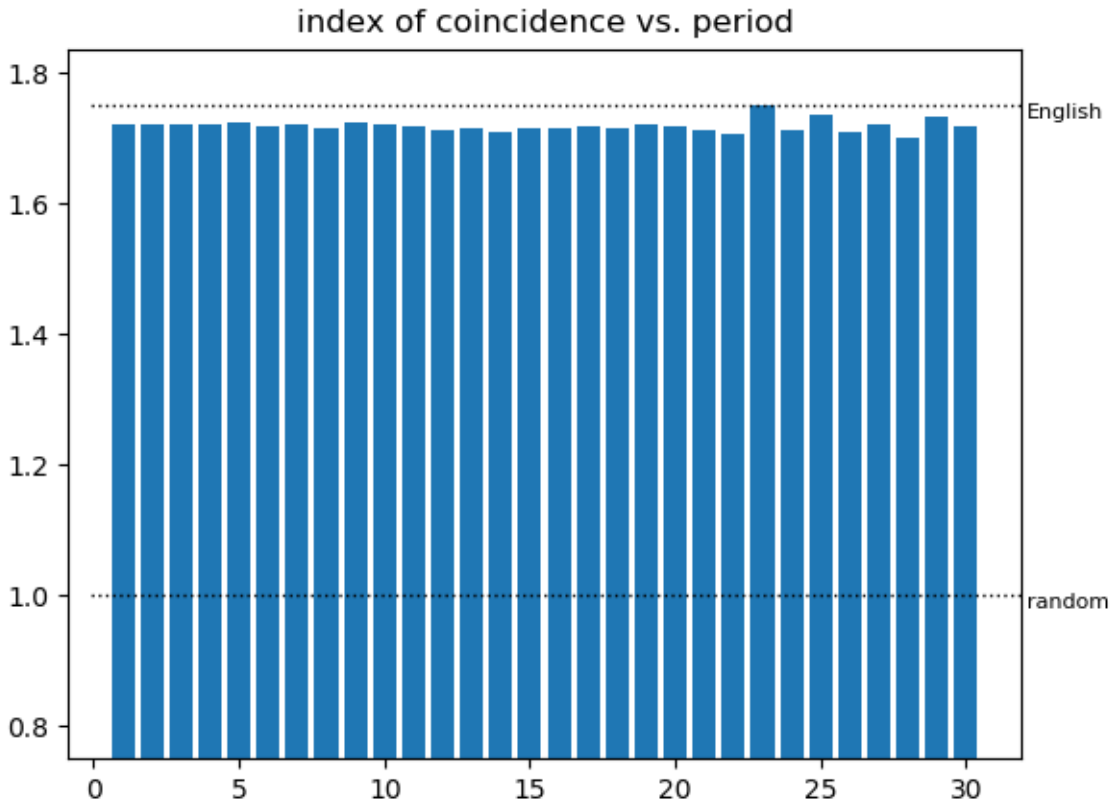
I process the entire ciphertext in this way and constructed an intermediate ciphertext.

```
UQUDREDUPDATFVSUNVDQIJEQDRRFAQHDNGHQGHRCFSBHGWNAQGMGHQUGEQNGPDATFVSAD
CCNEHQGHGWEQCFUFVNEMHQGHFVSDRHBQPDATFVSADCCNEGMGHQUBNGEQJQNCQPNWDHQVH
FNCCMGQEFDIGRCNKFVHBQPQGFSVKBFABUIGHOQNP ...
```

Now that I had something that contained only letters, I could try the usual tools. First, we can look at the monogram (single-letter) frequencies:



They look like the shuffled frequencies of typical English text: one around 12% for “E”, a few around 7% for “N” and “I” and “O” etc., and a few very low for “X” and “Z” and maybe “Q” and “J”. The index of coincidence as a function of period (i.e., looking at the IoC for every letter, then every second letter, then every third letter, ...) was flat. Both graphs indicate a monoalphabetic substitution cipher.



The standard method for breaking a monoalphabetic substitution cipher comes to us from Jakobsen's 1995 paper in *Cryptologia*. It involves a stochastic program that maximizes a textual fitness based on sets of letters. For my implementation, I used a fitness based on tetragram (four-letter) frequencies and calculate it as the frequency of a tetragram from the text times the logarithm of the frequency of the same tetragram from typical English text. (If you compare that to Shannon's definition of entropy, there is a minus sign difference.) In my program, the key is an alphabet of 26 letters in some order. Two of the letters are chosen at random and swapped. If the plaintext that results has a better fitness than before the swap, then it continues with the new key; otherwise, it tries a different swap. This continues until the fitness does not improve for several thousand swaps. At that point, the program believes that it has found the correct plaintext.

I ran my program on the intermediate ciphertext, and the resulting text was the correct plaintext.

```
MEMOFROMDOCKINGMANOEUVREOFFICETOASTESTFLIGHTSPACESYSTEMSREASDOCKI
NGCOLLARTESTSPRELIMINARYTESTINGOFTHEDOCKINGCOLLARSYSTEMHASREVEALE
DAPOTENTIALLYSERIOUSFLAWINTHEDESIGNWHICHMUSTBEADDRESSEDBEFORETHES
YSTEMCANBECERTIFIEDTHEISSUECONCERNSTHEEMERGENCYRELEASESYSTEMBACKG
ROUNDTHEDOCKINGSY...
```

The key that the program found was

```
NOAPQRSBFYTCUVDWXEGHIJKLMZ
```

However, this is not the key intended by the challenge creators. Remember that I assigned letters to the numbers first, then decrypted as a substitution cipher. To get the key that goes directly from the ciphertext to the plaintext, we need the inverse of my program's key, which is

```
CHLORISTUVWXYABDEFGKMNPQJZ
```

This key is more orderly than the other, and it fits with the clues about mythology from the text of challenge 9A. Chloris was, in Greek mythology, a flower nymph who comforted the souls of the dead.

The true assignment of letters from numbers is thus this:

0000 = 0 → C	0110 = 8 → U	1020 = 16 → E
0001 = 1 → H	0111 = 9 → U	1021 = 17 → F
0010 = 2 → L	0120 = 10 → W	1100 = 18 → G
0011 = 3 → O	0121 = 11 → X	1101 = 19 → K
0020 = 4 → R	1000 = 12 → Y	1110 = 20 → M
0021 = 5 → I	1001 = 13 → A	1111 = 21 → N
0100 = 6 → S	1010 = 14 → B	1120 = 22 → P
0101 = 7 → T	1011 = 15 → D	1121 = 23 → Q

The full plaintext is presented in the Appendix.

Appendix: Plaintext for 9A

Here is the plaintext for challenge 9A. Errors in punctuation are mine. British spelling is Harry's.

Harry,

With the launch of the Apollo-Soyuz mission coming up I thought it would be a good idea to check in with Mike and make sure he didn't have any left-over surprises for us. After the tank-pressurization problem on Apollo XVII, I was not completely confident that he couldn't have set something in motion back when he was still working at NASA. He seemed pleased to see me—I don't think he gets many visitors—but we didn't have a lot to talk about, and it didn't take him long to figure out why I was there. He still claims he had nothing to do with what he called "the accidents", though he admits he sent the letter to the press. I pressed harder and asked him about the manifesto. It had been read out in court, so he knew we knew about it, but he was definitely trying hard to steer me off the topic, and I got the feeling he was hiding something.

I didn't get anything more out of the interview, but I continued to worry about it on the way back to headquarters, so when I got there I dug out the manifesto and took another look. I don't know how we missed it, but we never carried out forensics on the document. I guess I was so taken up with cracking the cipher and analysing its content that I forgot to ask someone to check it. As soon as I realised the mistake I sent it over to Langley and got them to run the tests. The results are conclusive: The manifesto was not produced on Mike's typewriter. I looked through the files but couldn't find anything written by him on another typewriter, even at work. There he had a secretary to type for him, and in any case the ink and typeface on the manifesto don't match any of the NASA machines either. A typewriter is too big and bulky to hide, and we didn't give Mike time to dispose of one, so I don't think the manifesto was written by him.

I read it again more carefully for clues. I should have noticed the phrase "our son" at the top. It would have reminded me that Mike was married. Putting that together with the keyword for the manifesto cipher "NIOBE", things began to get a little clearer. Homer wrote about Niobe in the Iliad. She was famous as the queen whose sons were killed by the gods in revenge for her pride. So I got in touch with Interpol and made enquiries about Mike's wife. Mike's wife is also English, and the manifesto uses the English spelling of "programme". Her friends say she had a breakdown after her son died, and when Mike moved back to the US she went to live with her in-laws. After what seemed like a slow recovery, she told them she was moving back to the States to be near her original family. That was eleven years ago, but her relatives stateside say they have not seen her, and no one knows where she went. I spoke to the security service in the UK, and they told me she was an engineer who was quite capable of carrying out the sort of sabotage we saw on the earlier Apollo flights. Significantly, she learned her trade working with the UK atomic-weapons authority AWRE, which fits perfectly with the worries in the manifesto. She left after her son died in Korea and started meeting with peace campaigners. Her security clearance was revoked, and soon after that she left the UK.

Soon after I received the report from London I took a call from the Apollo-Soyuz team at NASA. Someone reminded them about the problems we had with the lunar flights, so they called me back into audit their security. I found a trail of edited service records that reminded me of the tampered files associated with Apollo XIII. The files were all scrambled using keys that came from Homer, Apollodorus, and other Greek classics. So far we have cracked all but one of them and found and fixed

problems with guidance control, life support, and electrical power. But there is one file I can't crack. Without it, we don't know if there is some other critical system that Mike’s wife may have tampered with. She failed to provoke America into declaring war on Russia three years ago, but the peace is still far from easy. If anything happens to the crew of the ASTF we might find ourselves on the brink of war again. With Stafford, Brand, and Slayton scheduled for lift-off in the next few weeks, it is crucial that we break this cipher.

Appendix: Plaintext for 9B

Here follows the plaintext. I punctuated and capitalized as best I could. When the official plaintext is released, we can see how the challenge creators meant to render it.

MEMO
From: Docking-manoeuvre office
To: AS test-flight space systems
RE: AS docking collar tests

Preliminary testing of the docking collar system has revealed a potentially serious flaw in the design which must be addressed before the system can be certified. The issue concerns the emergency release system.

Background: The docking system performs the following functions: impact energy absorption, mechanical connection, spacecraft alignment, and retraction, spacecraft hard mechanical connection, and docking interface sealing, spacecraft undocking, and separation. In order to achieve these functions, the docking system consists of three principle parts: the base, a structural ring, and the latching ring. The docking system base is the main structural member to which the docking system assemblies are attached. The structural ring carries the body latches, which provide a hard pressure-tight connection between the two spacecraft. Together with the capture latches which operate during the docking manoeuvre, these perform the docking function. They consist of eight active and eight passive hooks with an electrical drive installed on one of the latches and closed-loop cables connecting them. Each active hook has a cam-operated mechanism which performs its opening and tightening. Corresponding hooks of the passive docking system are captured by active hooks. Each passive hook has a stack of preloaded bellville springs providing a definite force for the docking interface. Pre-loading the docking interface seal will provide pressure integrity of the docking interfaces. This consists of two concentric rubber ring seals on each system, and a manhole cover is used to close the transfer tunnel of the spacecraft. Hatch locking and unlocking is manually performed by the crew. It is sealed by a mechanism which has a further eight eccentric latches, these being connected with each other by means of closed-cable connection. The docking system is equipped with alarm and meter system which provide telemetry to the ships and to ground control. In standard operational mode undocking is performed by release of the active spacecraft capture latches and then by opening the structure latch hooks. If necessary, undocking can be performed by the passive spacecraft by releasing the body-

mounted latches and opening the structure latch passive hooks. Spacecraft separation is performed by spring thrusters symmetrically located on the structural rings of both systems. After the latches release, the principle difference between the Russian and US docking system designs can be seen in the guide-ring system. Unlike the Russian electro-mechanical docking system, Apollo is equipped with an electric drive which uses cable connections to trigger the latches. Another essential difference is the Russian emergency release system (ERS), a backup provided by pyro bolts attached to each passive and active hook which operates in passive mode and provides practically instantaneous undocking in the event of a system malfunction or accident onboard one or both of the docked spacecraft. Situations in which the ERS might be initiated include

I. Uncontrolled fire or explosion onboard one of the spacecraft during docked operations;
II. Failure of the docking control system, preventing standard release operation;
III. Attitude control failure of, or unplanned firing on, one or both spacecraft, imposing high stresses on the docking mechanism.

While the ERS provides an effective backup for emergency situations, simulations and tests on the Huntsville docking test bed show that the effectiveness of the pyrotechnic bolts is a critical issue. Too much explosive could cause critical damage to the pressure seals around the hatch, while too little can leave the spacecraft attached with a damaged mechanism. The ERS was designed for operation on USSR missions and is therefore tuned to the structural constraints on the Soviet platform. Since the latches are electromagnetic on that spacecraft, they are less prone to damage under vibrational forces. The motor and cable mechanism used on the Apollo platform is more vulnerable to shock, and under certain test conditions has been shown to fail, following the triggering of the ERS on the Soviet end of the docking mechanism. Shockwaves will not, of course, propagate through the vacuum of space, but the forces can be transmitted through the tunnel to the Apollo latches, and if the bolts fire asymmetrically this places a torsion loading on the mechanism which can unseat the drive cables. In five of the seven tests where this phenomenon was observed the engineers were able to reseat the cables by repeatedly operating the mechanism, but in the remaining two cases the mechanism was beyond repair without manual intervention. Unfortunately, in these two cases the safety interlock also prevented the hatch from being opened, which could make it difficult for the astronauts to carryout a spacewalk to execute the required repair. In this case, it would normally be possible for the astronauts to manually operate the manhole latches by

disassembling and subsequently assembling the hatch cover. However, if the latching mechanisms have been sufficiently damaged by the pyro bolts this might prove a risky option, and it is even possible that the cover latches would fail to retract manually.

We have asked Queen's team to take a look at this, and she assures me that they can sort it out. She has worked on most of the Apollo mission design teams, so I am pretty confident that she can make sure everything is OK for this one.