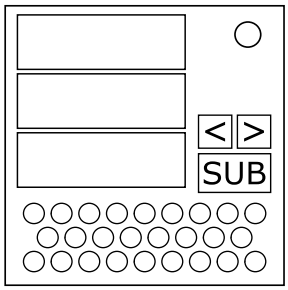
On the Subject of Arithmetic Ciphers



Death to software patents!

This module contains 3 screens, a keyboard, 2 arrows, and a submit button that displays the current page you're on.

Pressing the left or right arrow takes you to the previous or next page. There are 2 pages.

To disarm the module, decrypt a word using the following three steps. Once you have the decrypted word, type it in using the keyboard. When you start typing, the screens go blank and the bottom screen will show what you are typing.

To clear your input, click one of the arrows.

Once you are satisfied with your input, press the button labeled "SUB" to submit your answer.

A	00000
B	00001
C	00010
D	00011
E	00100
F	00101
G	00110
H	00111
I	01000
J	01001
K	01010
L	01011
M	01100
N	01101
O	01110
P	01111
Q	10000
R	10001
S	10010
T	10011
U	1010
V	1011
W	1100
X	1101
Y	1110
Z	1111

Step 1: Encrypted Binary Retrieval

For this step, use the letters from the top screen on page 1. Convert this encoded string to binary by replacing each letter with a binary code from the table on the right.

This will be referred to as the binary stream. Bits are extracted and removed from it from left to right.

Step 2: Frequency Retrieval

Concatenate the letters on the middle and bottom screens on page 1 and all three screens on page 2. Replace each letter with its position in the alphabet to obtain 27 numbers in the range 1-26. Label the first 26 with the letters A-Z and the last as "EOF". These numbers are referred to as frequencies.

For convenience with the following steps, also calculate the running total for each entry. Label A with 0, B with the frequency of A, C with the frequency of A+B, etc., until EOF with the sum of the frequencies of all letters A to Z.

Also calculate a value called *total* which is the sum of all frequencies (or, equivalently, the running total of EOF plus its frequency).

Step 3: Arithmetic Decryption

Throughout the process, three 20-bit values are relevant:

- *high* — starts out as 1048575 (or 11111111111111111111 in binary).
- *low* — starts out as 0 (or 00000000000000000000 in binary).
- *code* — starts out with a value obtained by extracting and removing the first 20 bits from the binary stream.

Perform the following steps repeatedly:

- All divisions in the following formulas are integer division (round down).
- Calculate the value of ((code − low + 1) * total − 1) / (high − low + 1).
- If the running total for EOF is ≤ this value, you are done.
- Otherwise, find the latest letter whose running total is ≤ to this value.
- Add this letter to your solution word.
- Calculate new values for *high* and *low* as follows:
  - high = (oldhigh − oldlow + 1) * (v + f) / total + oldlow − 1
  - low = (oldhigh − oldlow + 1) * v / total + oldlow

  where *v* is the running total of the letter just decoded, and *f* is its frequency.
- Convert *high*, *code* and *low* to 20-digit binary (with leading zeros if necessary) and modify them as follows:
  - Starting at the left, remove a contiguous block of bits where *high* and *low* are equal (marked in red below).
  - Keep the next column, then remove a contiguous block of bits where *high* is 0 and *low* is 1 (marked in blue below).
  - Pad *high* on the right back to 20 bits using only 1-bits.
  - Pad *low* on the right back to 20 bits using only 0-bits.
  - Pad *code* on the right back to 20 bits by extracting and removing bits from the binary stream. Use 1-bits if you exhaust the binary stream.