Binary Decoder

Write a **Go language** program that can decode various binary encoded messages. This is a group programming assignment (i.e., **only one submission per team is needed**).

Notes and Requirements:

- Submit your source code module only (I will provide my own binary encoded messages to test with). Your source code module should be a zipped folder containing at least
  - your Go source code (e.g. `binary.go`)
  - your module file (i.e. `go.mod`)
- Read the binary encoded message from `stdin`;
- Send the decoded output to `stdout`;
- Binary input may either be 7- or 8-bit ASCII (which you can automatically detect – or just output both);
- Original ASCII input will only contain "printable" characters;
- Therefore, whitespace characters (e.g., space, tab, carriage return, linefeed, etc) are acceptable (and you should replicate them in the output); and
- To make this even more interesting, backspaces may also be included in the original ASCII input (which, of course, you should "replicate" in the output)!

Please, no GUIs. Make this a command line application without frills that I can execute at the command line as illustrated below. Here are several runs of my program on various inputs:

```
binary1.txt
100100011001011101100110110011011110100000101011111011111110010110110011001
000100001

prof@latech:~$ ./binary < binary1.txt
Hello World!

binary2.txt
010010000110010101101100011011000110111100100000010101110110111101110010011
011000110010000100001

prof@latech:~$ ./binary < binary2.txt
Hello World!
```

```
tricky.txt
1110101111001111001011110010011101011100101101111110111111101000010000011100
0011000011110011111001101110101100001111001111101001110010110111111101110110
1111110110111110010001000110010111110010

prof@latech:~$ ./binary < tricky.txt
user:root pass:astronomer
```

Let's look at the last example a bit more closely:

```
1110101 1110011 1100101 1110010 0111010 1110010 1101111 1101111 1110100
  117     115     101     114     58      114     111     111     116
   u       s       e       r      :        r       o       o       t
0100000 1110000 1100001 1110011 1110011 0111010 1100001 1110011 1110100
  32      112     97      115     115     58      97      115     116
 [sp]      p       a       s       s      :        a       s       t
1110010 1101111 1101110 1101111 1101101 1111001 0001000 1100101 1110010
  114     111     110     111     109     121      8      101     114
   r       o       n       o       m       y     [bs]     e       r
```

First, notice that it is encoded in 7-bit ASCII. Second, it contains a backspace (ASCII 8). This is rather unfortunate. It's as if someone accidentally typed "astronomy" as a password, and fixed it by backspacing over the "y" and adding "er" to change it to "astronomer." Imagine such a sly trick during Cyber Storm...