# CCO '06 P3 - Codec

**Time limit:** 2.0s **Memory limit:** 256M

### Canadian Computing Competition: 2006 Stage 2, Day 1, Problem 3

**Note: The problem has been modified from the original so that it can work under the DM::OJ system. For the original problem statement, click here.**

The problem of lossless data compression is to transform some data into a compressed form such that:

- The original can be reproduced exactly from the compressed form
- The compressed form is as small as can reasonably be achieved

You are to write a program that first encodes a line of input into a string of bits, and then is able to decode that string of bits back into the original line of input. Your program will be graded on both its correctness and the length of the encoded string it outputs. Additionally, your program will be run twice, first with the original input (you are expected to output the encoded string) and then with the encoded string you outputted from the first run (you are expected to output the original input text.

Scoring is as follows:

- If you do not output the original text on the second run of your program, you score $0$ points
- If the encoded string consists of anything other than the characters `1` and `0`, you score $0$ points
- If the encoded string has a length **strictly greater than** $8$ times that of the original input text, you score $0$ points
- Otherwise, you score

  $$\min\left(50 \times \sqrt{8 - \frac{|E|}{|S|}}, 100\right)$$

  out of a possible $100$ points (where $|S|$ is the length of the original input text and $|E|$ is the length of your encoded string).

## Constraints

$1 \le |S| \le 10^6$, where $|S|$ is the length of the original input text.

## Input Specification

The first line contains a single integer $T \in \{1, 2\}$. If $T = 1$, then it's the first run of your program; otherwise, it's the second run of your program.

If $T = 1$, the next lines of input will contain the original input text. The data to compress will be English text represented by ASCII-printable characters (ASCII values between $32$ and $126$ **inclusive** or the newline character with ASCII value $10$).

If $T = 2$, the second line of input will contain your output from the first run. It will consist of only the characters `0` and `1`, terminated by a newline character (ASCII value $10$).

## Output Specification

If $T = 1$, output the encoded string, consisting of only the characters `0` and `1`, terminated by a newline character (ASCII value $10$).

If $T = 2$, output the original input text.

**Note that your output when $T = 2$ must match the original input text exactly, including whitespace.**

## Sample Input 1 (First Run)

```
1
Hello, World!
Goodbye, World!
```

## Sample Output 1 (First Run)

```
010101010
```

## Sample Input 2 (Second Run)

```
2
010101010
```

## Sample Output 2 (Second Run)

```
Hello, World!
Goodbye, World!
```