


# Problem F

## Erase Securely

**Problem ID:** erase  
**CPU Time limit:** 1 second  
**Memory limit:** 1024 MB

**Authors:** Ulf Lundström and  
Andreas Schuh

**Source:** Nordic Collegiate  
Programming Contest (NCPC),  
2013

**License:** 

Jon Marius is the computer expert at his company and has now been tasked to find some software for erasing data properly. It is very important that the data should not be recoverable afterwards, so it should be overwritten on the hard drive several times. Unable to find any free program up to the task, Jon Marius decides to write such a program himself. The user interface is simple, it only asks for the file to be destroyed and *[Math Processing Error]*, the number of times it should be overwritten. This number can range from *[Math Processing Error]* (quick deletion) to *[Math Processing Error]* (maximum security). Jon Marius processes the file bit by bit and does not consider writing a zero where there was already a zero as really overwriting. So for each of the *[Math Processing Error]* sweeps, he overwrites each zero with a one, and each one with a zero, respectively.



Photo by Hometown Beauty

Jon Marius knows that independent testing is important, so he has asked you to write the verification routine. He will not listen to your objections to the algorithm so eventually you give in.

### Input

The first line of the input contains a single integer *[Math Processing Error]*. The two following lines each contain a string containing only the characters 0 and 1. The first of these lines represent the bits of the file before deletion and the second the bits on the same position on the hard drive after the file has been deleted. The length of these strings are the same and between *[Math Processing Error]* and *[Math Processing Error]* characters.

### Output

Output a single line containing either the words “Deletion succeeded” if each bit is switched *[Math Processing Error]* times or “Deletion failed” if this is not the case.

#### Sample Input 1

```
1
10001110101000001111010100001110
0111000101011110000101011110001
```

#### Sample Output 1

```
Deletion succeeded
```

#### Sample Input 2

```
20
0001100011001010
0001000011000100
```

#### Sample Output 2

```
Deletion failed
```