

# Locked Up

October 2022

C++ — 2 SEC — 512 MB

Although theft is uncommon within the Guild of Sorcerers (thankfully), the commuting witches often secure their broomsticks with state-of-the-art combination locks. As with all things witchy, these combination locks involve an eye of newt and a gallon of witchcraft.

Each combination lock is comprised of  $n$  magical number weevils. Due to obscure and bewitching regulations (GOSR 24432),  $n$  can only take the values  $3k$  and  $3k+1$ . A witch can change the value of a number weevil by casting spells. Red spells decrease the weevil's value, while blue spells increase it (each by 1). As the weevils are small, a spell targeting one weevil also affects its adjacent neighbours.

The number weevils, while enchanting, are incredibly stupid, only understanding the numbers 0-9. Thus, a spell cannot be cast that would make a weevil's number go below 0 or above 9.

At rush hour, the sky-traffic can stretch for miles, so every witch wants to leave as quickly as possible. Each witch desires to know the minimum number of spells they need to cast to unlock their brooms.

**INPUT** First you will receive an integer  $n$ , the number of weevils that make up the combination lock. Next, you will receive a string of  $n$  digits, representing the current combination of lock, followed by a second string of  $n$  digits, representing the combination required to open the lock.

$$1 \leq k \leq 1,000$$

**OUTPUT** Output a single integer, the minimum number of spells that need to be cast to convert the first combination into the second.

**SAMPLE** For example, suppose the combination lock has 4 weevils, starting with the combination 0000. The weevils are numbered 0-3 from left to right. Casting the spells 0B 2B 0B 3B 3B 1R 3B transforms the combination to 1234 via 1100 1211 2311 2322 2333 1223 1234. This is the minimum number of spells required to transform 0000 to 1234.

INPUT	OUTPUT
3 123 321	4
6 010101 234789	14
10 0918273645 1357908642	50