Problem 1 – Key Revolver

Our favorite super-spy action hero Sam is back from his mission in another exam, and this time he has an even more difficult task. He needs to **unlock a safe**. The problem is that the safe is **locked** by **several locks in a row**, which all have **varying sizes**.

Our hero posesses a special weapon though, called the **Key Revolver**, with special bullets. Each **bullet** can unlock a **lock** with a **size equal to or larger than** the **size** of the **bullet**. The bullet goes into the keyhole, then explodes, completely **destroying** it. Sam **doesn't know the size** of the locks, so he needs to just shoot at all of them, until the safe runs out of locks.

What's behind the safe, you ask? Well, intelligence! It is told that Sam's sworn enemy – **Nikoladze**, keeps his **top secret Georgian Chacha Brandy** recipe inside. It's valued differently across different times of the year, so Sam's boss will tell him what it's worth over the radio. One last thing, every bullet Sam fires will also cost him money, w**hich will be deducted from his pay** from the price of the intelligence.

Good luck, operative.

Input

- On the **first line** of input, you will receive the price of each **bullet** – an **integer in the range [0-100]**
- On the **second line**, you will receive the **size of the gun barrel** – an **integer in the range [1-5000]**
- On the **third line**, you will receive the **bullets** – a **space-separated integer sequence** with **[1-100] integers**
- On the **fourth line**, you will receive the **locks** – a **space-separated integer sequence** with **[1-100] integers**
- On the **fifth line**, you will receive the **value of the intelligence** – an **integer in the range [1-100000]**

After Sam receives all of his information and gear (**input**), he starts to **shoot the locks front-to-back**, while going through the bullets **back-to-front**.

If the **bullet** has a **smaller or equal** size to the **current lock**, print "**`Bang!`**", then **remove the lock**. If not, print "**`Ping!`**", leaving the lock **intact**. The bullet is removed in **both cases**.

If Sam runs out of bullets in his barrel, print "**`Reloading!`**" on the console, then continue shooting. If there aren't any bullets left, **don't** print it.

The program ends when Sam **either runs out of bullets**, or the safe **runs out of locks**.

Output

- If Sam **runs out of bullets** before the safe runs out of **locks**, print:
  "**`Couldn't get through. Locks left: {locksLeft}`**"
- If Sam manages to **open the safe**, print:
  "**`{bulletsLeft} bullets left. Earned ${moneyEarned}`**"

Make sure to account for the **price of the bullets** when calculating the **money earned**.

Constraints

- The input will be **within the constaints** specified above and will **always be valid**. There is **no need** to check it explicitly.
- There will **never** be a case where Sam breaks the lock and ends up with **negative balance**.

Examples

Input	Output	Comments

50
2
11 10 5 11 10 20
15 13 16
1500
	Ping!
	Bang!
	Reloading!
	Bang!
	Bang!
	Reloading!
	2 bullets left. Earned $1300
		20 shoots lock **15** (**ping**)
		10 shoots lock **15** (**bang**)
		11 shoots lock **13** (**bang**)
		5 shoots lock **16** (**bang**)
		
		Bullet cost: 4 * 50 = $200
		Earned: 1500 – 200 = $1300

20
6
14 13 12 11 10 5
13 3 11 10
800
	Bang!
	Ping!
	Ping!
	Ping!
	Ping!
	Ping!
	Couldn't get through. Locks left: 3
		5 shoots lock **13** (**bang**)
		10 shoots lock **3** (**ping**)
		11 shoots lock **3** (**ping**)
		12 shoots lock **3** (**ping**)
		13 shoots lock **3** (**ping**)
		14 shoots lock **3** (**ping**)

33
1
12 11 10
10 20 30
100
	Bang!
	Reloading!
	Bang!
	Reloading!
	Bang!
	0 bullets left. Earned $1
		10 shoots lock **10** (**bang**)
		11 shoots lock **20** (**bang**)
		12 shoots lock **30** (**bang**)
		
		Bullet cost: 3 * 33 = $99
		Earned: 100 – 99 = $1