Problem I. Telepathy

Input file:	*standard input*
Output file:	*standard output*
Time limit:	2 seconds
Memory limit:	512 mebibytes

Brothers Flim and Flam are performing a trick they call "telepathy".

At the beginning, Discord, who is the host, generates two random binary strings a and b. Each string contains $n = 10^6$ digits, and each digit is equal to zero or one equiprobably and independently of other digits. String a is given to Flim, and string b to Flam. Each of them sees only his own string, and doesn't know the string of his brother.

After that, each brother selects $k = 10^5$ distinct positions in the string: not in his own, but in his brother's string that they do not know!

Finally, Discord looks at string a from left to right, and writes down the digits from the positions selected by Flam. Then he looks at string b from left to right and, under the previous line, writes down the digits from the positions selected by Flim. After that, the audience counts how many times a digit from a turned out to be the same as a digit from b written under it. To "prove" that telepathy works, **more than two thirds** of the pairs of digits have to turn out the same, that is, at least 66 667 of them.

Help Flim and Flam to plan how to select positions in each other's strings knowing only their own string, so that they can "prove" that telepathy works.

Consider a small example.

- To keep things short, let $n = 20$ and $k = 5$.
- Let string a be `00101011011110111001`.
- Let string b be `11000111101000011010`.
- Flim sees string a and selects positions $2, 3, 5, 7, 11$ in string b.
- Flam sees string b and selects positions $1, 4, 9, 16, 20$ in string a.
- Discord writes down $a_1 = 0$, $a_4 = 0$, $a_9 = 0$, $a_{16} = 1$, $a_{20} = 1$.
- And under them, $b_2 = 1$, $b_3 = 0$, $b_5 = 0$, $b_7 = 1$, $b_{11} = 1$.
- Out of five pairs, the digits are the same in each pair except the first ($a_1 = 0$ but $b_2 = 1$).
- It means Flim and Flam achieved $4/5$ equalities.
- The portion of equalities is greater than $2/3$, so the brothers "proved" that telepathy works!

Interaction Protocol

In this problem, your solution will be run twice on each test: acting as Flim during the first run and acting as Flam during the second run. The jury program is acting for Discord. Each line of input is terminated by an end-of-line character.

First Run

During the first run, the solution acts for Flim. The first line contains the name "`Flim`". The second line contains two space-separated integers n and k: the length of the string and the number of positions to select (in all tests of the problem, $n = 10^6$ and $k = 10^5$). The third line contains n binary digits without spaces: the string a that is given to Flim.

Print k space-separated integers $1 \le p_1 < p_2 < \ldots < p_k \le n$: the selected positions in string b.

Second Run

During the second run, the solution acts for Flam. The first line contains the name "`Flam`". The second line contains two space-separated integers n and k: the length of the string and the number of positions to select (here, again, $n = 10^6$ and $k = 10^5$). The third line contains n binary digits without spaces: the string b that is given to Flam.

Print k space-separated integers $1 \le q_1 < q_2 < \ldots < q_k \le n$: the selected positions in string a.

After the second run, the jury program compares the digits in k pairs: first a_{q_1} with b_{p_1}, then a_{q_2} with b_{p_2}, and so on. If the equality holds in more than $\frac{2}{3}k$ of these pairs, the solution gets `OK` for the test. Otherwise, the outcome is `Wrong Answer`.

In this problem, there are one downloadable example and 50 secret tests. All binary strings are generated in advance by a pseudorandom number generator: each digit of each string is either zero or one, equiprobably and independently of other digits.

Example

Below we show two runs of a certain solution on the first test. The strings and answers are shown only partially for brevity. The full version of the example can be seen in `samples.zip`. There are 66 859 equal pairs in the example.

standard input	*standard output*

`Flim`
`1000000 100000`
`110111111110<...>11010111`
	`3 14 25 <...> 999979 999990`

standard input	*standard output*

`Flam`
`1000000 100000`
`000000110100<...>10011111`
	`7 16 21 <...> 999977 999992`