decrypt

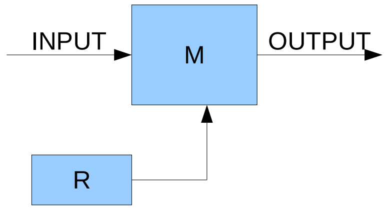
100 points

Source code: `decrypt.c`, `decrypt.cpp`, `decrypt.pas`

Time limit: **`0.1 seconds`**

Memory limit: **`64MB`**

There is a rumor that the Scientific Committee is using a special device for encrypting their communication. If you can crack the encryption you could listen to the problems they have prepared and score many points. Last night you got lucky: one of the members forgot their device in a bar. You opened it and looked at the general design:



All the operations use 8 bits. XOR is the bitwise exclusive or function (the **^** operator in C, **`xor`** operator in Pascal).

R is a sequence of pseudorandom numbers defined as follows:

- R[0], R[1] and R[2] have secret values only known by the Scientific Committee.
- For n = 3,4,… let R[n] = R[n-2] XOR R[n-3].

Furthermore, we have a function M, M:[0..255] → [0..255]. In fact, M is a bijection, i.e. when x≠y it must also be that M[x]≠M[y].

The device used by the Scientific Committee takes one number at a time, and outputs it encrypted.

After using the device N times, the next number, INPUT is encoded as follows:

- OUTPUT = M(INPUT XOR R[N])

Even though you understand how the encryption device works, you do not know the secret values R[0], R[1], and R[2]. Also, you do not know M, so you cannot decode the communication. What you can do instead is play with the device you found. You can give it input numbers and observe the outputs.

Task

Your task is to find out all the secret values: R[0], R[1], R[2], M[0], M[1], .., M[255] with less than **320** queries (input numbers).

Constraints

- A correct solution receives points only if the number of queries is less than 320.
- In all tests the secret keys R[0], R[1], R[2], M[0], M[1], …, M[255] are random.
- **`0 ≤ INPUT, OUTPUT, R, M ≤ 255`**

Interaction

This is an interactive program. To play with the encryption module simply write a number between 0 and 255 to the standard output. You can then read from the standard input the output of the encryption device, also an integer between 0 and 255. When you know the secret values output one line containing the string **SOLUTION** and after that output 259 lines: R[0], R[1], R[2], M[0], M[1], …, M[255], one value per line.

Programming instructions

After every complete line written to the standard output, C programmers must use fflush(stdout) function while Pascal programmers must use flush(output) procedure.

C	*C++*	*Pascal*
`printf("%d\n", q);` `fflush(stdout);`	`cout<<q<< '\n';` `cout.flush();`	`writeln(q);` `flush(output);`

Example

Let's assume that R[0] = 0, R[1] = 1, R[2] = 3 and that M[i] = (i + 1) MODULO 256.
From here we deduce R[3] = 1.

Contestant output	*Contestant input*	*Comments*
`10`	`11`	`M[10 XOR 0] = M[10] = 11`
`10`	`12`	`M[10 XOR 1] = M[11] = 12`
`11`	`9`	`M[11 XOR 3] = M[8] = 9`
`12`	`14`	`M[12 XOR 1] = M[13] = 14`
…		

`SOLUTION`
`0`
`1`
`3`
`1`
`2`
`3`
`4`
`…`
`254`
`255`
`0`
		When you find out the secret values you output them.