Task A1. TELEPORT

The software company, where the young programmer Stancho was hired, is famous with its distributed software for management of small planetary systems, the number of planets of which is a power of two. Recently, the company issued the current version of the software and the young professionals of the company have to visit the clients and to install the updates. Stancho also has to visit the planets of one system-customer. Traveling from one planet to another planet of the system is not easy. Imagine that the planets are ordered in line by their distances to the star of the system and labeled with the numbers $0, 1, \ldots, 2^n-1$ following this order (astronomically this is rare, but not impossible). Stancho has 2^n-1 teleports, labeled with $1, 2,\ldots,2^n-1$. The teleport labeled with t could transmit only once one human from a planet labeled with m to the planets with label $m+t$ or label $m-t$, if such a planet exists (i.e. if $0 \leq m-t$ and $m+t \leq 2^n-1$). Using a space-stop Stancho could go to the planet labeled with k. Write a program **teleport**, which for given n and k has to find the best way Stancho can use the teleports in order to visit as many as possible planets of the system.

Input

On the single line of the standard input the positive integers, n and k are given, $1 \leq n \leq 20, 0 \leq k \leq 2^n-1$.

Output

On the first line of the standard output the program has to print the maximal number p of planets that Stancho could visit – excluding the planet k. On the next row, the program has to print a sequence of the numbers of p teleports that Stancho has to use, in the order of using. The numbers of teleports that transmit him from a planet with larger number to a planet with smaller number has to be multiplied by –1. If there are several different ways to do the teleporting the program has to print any one of them.

Constraints

$1 \leq n \leq 20, 0 \leq k \leq 2^n-1$.

Example

Input	Possible output
2 2	3
	-1 2 -3