

Right rotation

The integer $(19)_{10}$ (decimal representation) is represented as $(10011)_2$ in the binary numeral system. This allows us to define the *rotation to the right* of a natural number as the operation that places the last digit of the binary representation before the first one. If we thus rotate the binary number $(10011)_2$ to the right, we obtain the binary number $(11001)_2 = (25)_{10}$. A given natural number can also be rotated to the right n -fold. In this case n successive rotations are executed on the binary representation of the number. As such, the 4-fold rotation to the right of 19 equals 7, as can be seen from the following series of rotations.

19	$\xrightarrow{\text{r}}$	25	$\xrightarrow{\text{r}}$	28	$\xrightarrow{\text{r}}$	14	$\xrightarrow{\text{r}}$	7
10011	$\xrightarrow{\text{r}}$	11001	$\xrightarrow{\text{r}}$	11100	$\xrightarrow{\text{r}}$	01110	$\xrightarrow{\text{r}}$	00111

If the binary representation of an integer has leading zeroes after a (n -fold) rotation to the right, the leading zeroes are deleted.

The inverted operation that places the first digit of the binary representation of an integer at the back, is logically called a *rotation to the left* of the integer. Analogous to a rotation to the right, an n -fold rotation to the left is defined as a repeated execution of a rotation to the left. Hence, the 4-fold rotation to the left of the number 357 equals 91, as can be seen from the following series of rotations.

357	$\xrightarrow{\text{l}}$	203	$\xrightarrow{\text{l}}$	406	$\xrightarrow{\text{l}}$	301	$\xrightarrow{\text{l}}$	91
101100101	$\xrightarrow{\text{l}}$	011001011	$\xrightarrow{\text{l}}$	110010110	$\xrightarrow{\text{l}}$	100101101	$\xrightarrow{\text{l}}$	001011011

Input

The input exists of two lines, the first containing a number $g \in \mathbb{N}$, and the second a number $n \in \mathbb{N}$.

Output

Execute an n -fold rotation of the number g , and write out the result of all intermediate steps. Execute a rotation to the right if n is positive, and a rotation to the left if n is negative. The following intermediate results must be written out:

- the decimal representation g_{10} of the number g
- the binary representation g_2 of the number g
- the binary representation g_2^n of the number g after n rotations, with leading zeroes removed
- the decimal representation g_{10}^n of the number g after n rotations

Tip: use the built-in Python functions `bin` and `int` to convert the decimal representation of an integer to its binary representation and *vice versa*.

Example

Input:

```
19
4
```

Output:

```
19
10011
111
7
```

Example

Input

```
357
-4
```

Output:

```
357
101100101
1011011
91
```

Het natuurlijk getal $(19)_{10}$ (decimale voorstelling) kan geschreven worden als $(10011)_2$ in het binair talstelsel. Dit laat ons toe om de *rotatie naar rechts* van een natuurlijk getal te definiëren als de bewerking waarbij het laatste cijfer vooraan geplaatst wordt in de binaire voorstelling van het getal. Als we op die manier het binair getal $(10011)_2$ naar rechts roteren, dan bekomen we het binair getal $(11001)_2 = (25)_{10}$. Een gegeven natuurlijk getal kan ook n -voudig naar rechts geroteerd worden. In dat geval worden n opeenvolgende rotaties naar rechts uitgevoerd op de binaire voorstelling van het getal. De 4-voudige rechtse rotatie van het getal 19 is dan gelijk aan 7, zoals kan afgeleid worden uit de onderstaande rotatiereeks.

19	$\xrightarrow{\text{r}}$	25	$\xrightarrow{\text{r}}$	28	$\xrightarrow{\text{r}}$	14	$\xrightarrow{\text{r}}$	7
10011	$\xrightarrow{\text{r}}$	11001	$\xrightarrow{\text{r}}$	11100	$\xrightarrow{\text{r}}$	01110	$\xrightarrow{\text{r}}$	00111

Als na een (n -voudige) rotatie naar rechts een aantal nullen vooraan een binair getal komen te staan, dan mogen die weggelaten worden.

De omgekeerde bewerking, waarbij het eerste cijfer in de binaire voorstelling van een natuurlijk getal achteraan geplaatst wordt, wordt dan logischerwijs de *rotatie naar links* van het getal genoemd. Analoog als bij de rotatie naar rechts, wordt ook de n -voudige rotatie naar links gedefinieerd als het herhaald uitvoeren van de rotatie naar links. De 4-voudige linkse rotatie van het getal 357 is dan 91, zoals kan afgeleid worden uit de onderstaande rotatiereeks.

357	$\xrightarrow{\text{l}}$	203	$\xrightarrow{\text{l}}$	406	$\xrightarrow{\text{l}}$	301	$\xrightarrow{\text{l}}$	91
101100101	$\xrightarrow{\text{l}}$	011001011	$\xrightarrow{\text{l}}$	110010110	$\xrightarrow{\text{l}}$	100101101	$\xrightarrow{\text{l}}$	001011011

Invoer

De invoer bestaat uit twee regels, waarvan de eerste regel een getal $g \in \mathbb{N}$ bevat, en de tweede regel een getal $n \in \mathbb{N}$.

Uitvoer

Voer een n -voudige rotatie uit van het getal g , en schrijf alle tussenstappen uit. Indien n positief is moet een rotatie naar rechts uitgevoerd worden, voor negatieve n moet een rotatie naar links uitgevoerd worden. De tussenstappen die moeten uitgeschreven worden zijn:

- de decimale voorstelling g_{10} van het getal g
- de binaire voorstelling g_2 van het getal g
- de binaire voorstelling g_2^n van het getal g na n rotaties, waarbij voorloopnullen worden weggelaten
- de decimale voorstelling g_{10}^n van het getal g na n rotaties

Tip: ga na hoe de ingebouwde Python functies `bin` en `int` kunnen gebruikt worden om de decimale voorstelling van een natuurlijk getal om te zetten naar zijn binaire voorstelling, en omgekeerd.

Voorbeeld

Invoer:

```
19
4
```

Uitvoer:

```
19
10011
111
7
```

Voorbeeld

Invoer:

```
357
-4
```

Uitvoer:

```
357
101100101
1011011
91
```