## Problem 5 – Catch the Bits

You are given a **sequence of bytes**. Consider each byte as sequence of exactly 8 bits. You are given also a number **step**. Write a program to extract all the bits at positions: **1**, **1 + step**, **1 + 2\*step**, ... Print the output as a sequence of bytes. In case the last byte have less than 8 bits, add trailing zeroes at its right end. Bits in each byte are counted from the leftmost to the rightmost. Bits are numbered starting from 0.

### Input

* The input data should be read from the console.
* The number **n** stays at the first line.
* The number **step** stays at the second line.
* At each of the next **n** lines **n** bytes are given, each at a separate line.

The input data will always be valid and in the format described. There is no need to check it explicitly.

### Output

The output should be printed on the console. Print the output bytes, each at a separate line.

### Constraints

* The number **n** will be an **integer** number in the range [1…100].
* The number **step** will be an **integer** number in the range [1…20].
* The **n numbers** will be integers in the range [0…255].
* Allowed working time for your program: 0.25 seconds.
* Allowed memory: 16 MB.

### Examples

|  |  |  |
| --- | --- | --- |
| **Input** | **Output** | **Comments** |
| 2  11  109  87 | 128 | We have the following input sequence of 16 bits (2 bytes):  0**1**101101 0101**0**111.  We take the bits 1 and 12 (step=11). We obtain the sequence **10**.  We pad the sequence with 6 trailing zeroes. Result: **10000000**. |

|  |  |  |
| --- | --- | --- |
| **Input** | **Output** | **Comments** |
| 3  2  45  87  250 | 63  192 | We have the following input sequence of 24 bits (3 bytes):  0**0**1**0**1**1**0**1** 0**1**0**1**0**1**1**1** 1**1**1**1**1**0**1**0**  We take bits 1, 3, 5, …, 23 (step=2). We obtain the sequence:  00111111 1100. We pad it with 4 zeroes to obtain 2 full bytes. Result: **00111111 11000000**. |

|  |  |  |
| --- | --- | --- |
| **Input** | **Output** | **Comments** |
| 2  2  45  87 | 63 | We have the following input sequence of 16 bits (2 bytes):  0**0**1**0**1**1**0**1** 0**1**0**1**0**1**1**1**  We take bits 1, 3, 5, …, 15 (step=2). We obtain the sequence:  00111111 (8 bits). No padding is needed. Result: **00111111**. |