Decrypt me (decrypt)

Alice sends Bob a secret message, composed of n bytes indexed with an integer ranging from 0 to $n-1$.

She uses a simple encryption scheme that replaces every byte x of index i with a new value y, obtained as a right circular shift of x by (i mod 8) positions, where (i mod 8) denotes the remainder of the integer division of i by 8 (i % 8 in C).

For instance, if the byte x of index 2 of the message contains the character 'a' (hex 61 in ASCII, binary 01100001), the Alice's encryption replaces x with y obtained as right circular shift of 2 positions, that is, 01011000, hex 58.

You are to help Bob decrypt Alice's message. Write a program that takes as input an encrypted text and produces as output the decrypted text by reversing Alice's scheme as follows:

Byte y with index i in the encrypted input text is replaced by byte x, where x is obtained by a left circular shift of the bits of y by i mod 8 positions.

Example:

Encrypted input bytes (hex): 61 B0 58 2C 16 0B 85 C2 61

Decrypted output text (ascii): "aaaaaaaaa"

		input y (data)		output (x) data			
i	**i % 8**	**hex**	**binary**	**char**	**dec**	**hex**	**binary**
0	0	61	01100001	a	97	61	01100001
1	1	b0	10110000	a	97	61	01100001
2	2	58	01011000	a	97	61	01100001
3	3	2c	00101100	a	97	61	01100001
4	4	16	00010110	a	97	61	01100001
5	5	0b	00001011	a	97	61	01100001
6	6	85	10000101	a	97	61	01100001
7	7	c2	11000010	a	97	61	01100001
8	0	61	01100001	a	97	61	01100001

Implementation

You should submit a single file, with either a `.c`, `.cpp`, `.java` or `.py` extension.

Your program must read input data from `stdin` and write the output data into `stdout`.

`stdin` consists two lines:

- Line 1: The integer n, the number of bytes of the secret message.
- Line 2: n hex bytes, separated by space, representing the encrypted message

`stdout` consists of only one line:

- Line 1: The decrypted text (ASCII encoded).

Constraints

- $1 \le n \le 1024$.
- All the input hex bytes are well formed (two digits, lowercase, from 00 to **ff**)

Scoring

Your program will be tested against 10 testcases, each of which is worth 10 points.

Examples

stdin	stdout
`9` `61 b0 58 2c 16 0b 85 c2 61`	`aaaaaaaaa`