Reverse frequency scrambler (`reverse`)

Alice and Bob wish to make fun of their friends by letting they think they can fluently write to each other in some weird foreign language.

They want to use an algorithm A that can be applied both to scramble and to recover the original message, that is, for any given text t, A(A(t)) = t.

The algorithm should confuse the reader by leaving a certain amount of semi-intelligible meaning.

On a sunny terrace in Amsterdam, sipping a cup of tea, they think of a simple solution that works for any text on an alphabet of a given size, say 256 (well yeah, good ol' 8-bit ASCII).

Here's the idea. They take the text t and compute the frequency f(c) of each character c of the alphabet in t. For each distinct frequency value, they consider all characters having exactly that frequency and they sort them by ASCII code. In each resulting group, they exchange the ASCII codes of the smallest with the largest, the second smallest with the second largest, etc.

Example:

Given the text t = "EAEEABDBFBCDF", algorithm A obtains the frequency groups:

- $f('A') = f('D') = f('F') = 2$
- $f('B') = f('E') = 3$
- $f('C') = 1$
- All other frequencies are 0

Now, the transformation by reverse ASCII code will replace characters as follows:

- $A \to F$, $B \to E$, $C \to C$
- $D \to D$, $E \to B$
- $F \to A$,

Hence, t' = A(t) = "BFBBFEDEAECDA".

Since the transformation preserves the frequency of characters, the same algorithm A applied back to t' yields t = A(t') = "EAEEABDBFBCDF". Neat huh?

Write a program that, given an ASCII text t as input, produces the ASCII text t' = A(t) as output.

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 of only two line:

- Line 1: The integer n, the number of characters in the message t.

- Line 2: n ASCII characters, the message t.

`stdout` consists of only one line:

- Line 1: The decrypted message t'.

Constraints

- $1 \leq n \leq 100.000$.
- All the characters of the message t are printable.

Scoring

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

Examples

stdin	stdout
13 EAEEABDBFBCDF	BFBBFEFEAECDA