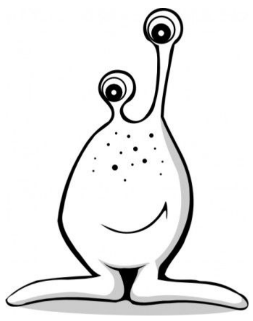
Task Ekoeko



You must be familiar with the story of an alien called Eko Eko, who got his name due to a malfunction in his translation device. The little alien is once again back on Earth to help earthlings with a clean-up after the Advent. However, Eko Eko's translation device has stopped working again.

This time, not only does the device repeat a word, but it also changes the order of the letters in a word. For example, the word "`slon`" first becomes "`slonslon`", and then by changing the order of the letters it could become "`slosnoln`" or "`soolnlsn`" etc. The amount of gold needed to repair Eoo Kek's device depends on the number of swaps of adjacent characters needed to make Kok Oee's badly translated word into a word which is made from just repetition.

For example, if Keo Koe's device translates a word to "`soolnlsn`", it is sufficient to make four swaps of adjacent characters to obtain a word which is made from a repetition - "`olsnolsn`" (see the clarification of the third example), so four pieces of gold are enough to repair his device. Notice that the word obtained from a sequence of such swaps is not necessarily the word that Koo Kee originally wanted to say. This does not effect the amount of gold needed to repair his device.

You would like to help Eke Koo, but if you steal a large amount of jewelry from your mother you won't get a Christmas present. That's why for a given word that came out of Keo Koe's translation device, you want to determine the least possible number of swaps of adjacent characters to get a word which is made from just repetition.

Input

The first line contains a positive integer n - the length of the word that Eek Ook is trying to say.

The second line contains a sequence of $2n$ characters, each being a lowercase letter of the Latin alphabet, representing the word that came out of Eok Eok's translation device. Each letter will appear an even number of times.

Output

In the only line print the least possible number of swaps of adjacent characters to make Keo Koe's word into a single-repetition word.

Scoring

Constraints on n are $1 \leq n \leq 100\ 000$ in all subtasks.

Subtask	Points	Constraints
1	10	The string consists of n occurrences of `a`, and then n of `b`.
2	20	Each letter appears at most twice.
3	20	The first n and the last n letters are the same, but possibly in a different order.
4	20	$1 \leq n \leq 1000$
5	40	No additional constraints.

Examples

input

```
3
koeeok
```

output

```
3
```

input

```
3
kekoeo
```

output

```
1
```

input

```
4
soolnlsn
```

output

```
4
```

Clarification of the third example:
One way to get from `soolnlsn` to a single-repetition word using four swaps is

$$\texttt{soolnlsn} \rightarrow \texttt{so}\underline{\texttt{lo}}\texttt{nlsn} \rightarrow \texttt{sol}\underline{\texttt{no}}\texttt{lsn} \rightarrow \underline{\texttt{os}}\texttt{lnolsn} \rightarrow \texttt{o}\underline{\texttt{ls}}\texttt{nolsn}$$