#A - KMP

Information

Time Limit	Memory Limit	Data Amount	Problem Type
1000-4000ms	128-512MiB	10	Tradition

Description

Give you a string, you should print the *Next*[] in the **KMP** algorithm.

Input

The first line contains one integer *n*, representing the length.

The second line contains a string of length *n*, only containing lowercase letters.

Output

n integers, representing the *Next*[] array.

Sample Test Data

Input #1

```
13
abcdabcabcdef
```

output #1

```
0 0 0 0 1 2 3 1 2 3 4 0 0
```

Tips

Data Limit

For 100% cases: 1<=n<=10^6, 1<=*n*<=10^6

Attention

#B - Picking foods

Description

There are *n* pieces of food in a row. Each piece of food is with or without cilantro. If a piece of food is without cilantro, Asuka will get 1 happiness after eating; Otherwise Asuka will get -1 happiness after eating(She doesn't like cilantro). Now she wants to eat foods arranging in a continuous interval, and she wants to get the maximum happiness. Please help her to calculate the maximum number.

Note that Asuka must eat at least one piece of food.

Formally, please calculate $max_{1\leq l\leq r\leq n}\{number\ of\ 0\ in[l,r]-number\ of\ 1\ in[l,r]\}$.

Input

The first line contains one integer *n*, representing the length.

The second line contains a string of length n*, only containing 0 and 1.

Output

One integer, representing the maximum happiness she can get.

Sample Test Data

Input #1

```
13
0010101011101
```

output #1

```
2
```

Tips

Data Limit

For 50% cases: $1<=n<=10^3$

For 100% cases: $1<=n<=10^6$

Attention

#C - String problem 3

Description

Peter has a string *A*. He repeats it to become *AA*, and inserts a character *c* in *AA* to get string *T*. You need to find the string *A*.

Input

The first line contains one integer *n*, representing the length.

The second line contains a string T*, containing only **uppercase** letters.

Output

If *A* does not exist, print "NOT POSSIBLE". If *A* is not unique, print "NOT UNIQUE". Otherwise, print *A*.

Sample Test Data

Input #1

```
7
JIKLJKL
```

output #1

```
JKL
```

Input #2

```
3
KMP
```

output #2

```
NOT POSSIBLE
```

Input #3

```
9
NBNBNBNBN
```

output #3

```
NOT UNIQUE
```

Data Limit

For 100% cases: $2 <= n <= 2 * 10^6 + 1$

#D - Recurring Period

Description

We take a string *B* and repeat it indefinitely, and then we take a certain prefix as string *A*.

Now we know *A* but not *B*, please calculate the minimum length of *B*.

Input

The first line contains one integer *n*, representing the length.

The second line contains a string *A*, containing only lowercase letters.

Output

One integer, representing the minimum length of *B*.

Sample Test Data

Input #1

```
8
cabcabca
```

output #1

```
3
```

Tips

We can repeat "cab" to form "cabcabcab", and "cabcabca" is its prefix.

Data Limit

For 50% cases: $1 <= n <= 10^3$

For 100% cases: $1 <= n <= 10^6$

Attention

#E - String problem 5

Description

Peter gives you 2 strings *A* and *P*. You should do the following operations.

1. If *P* is not in *A*, goto 4.
2. delete the first occurrence of *P* in *A*.
3. goto 1.
4. print *A*.
5. end.

Please use fast I/O in Java.

Input

The second line contains a string *A*, containing only **lowercase** letters.

The second line contains a string *P*, containing only **lowercase** letters.

Output

The string *A* after the operations.

Sample Test Data

Input #1

```
uujjklklpp
jkl
```

output #1

```
uupp
```

Input #2

```
uujjklklpp
ac
```

output #2

```
uujjklklpp
```

Data Limit

For 100% cases: the length of both *A* and *P* are less than 10^6.

Attention

It is guaranteed that *A* will not become empty during the operations.

Please use fast I/O in Java.