F. TorCoder

time limit per test: 3 seconds

memory limit per test: 256 megabytes

input: input.txt

output: output.txt

A boy named Leo doesn't miss a single TorCoder contest round. On the last TorCoder round number 100666 Leo stumbled over the following problem. He was given a string s, consisting of n lowercase English letters, and m queries. Each query is characterised by a pair of integers l_i, r_i $(1 \le l_i \le r_i \le n)$.

We'll consider the letters in the string numbered from 1 to n from left to right, that is, $s = s_1 s_2 \dots s_n$.

After each query he must swap letters with indexes from l_i to r_i inclusive in string s so as to make substring (l_i, r_i) a palindrome. If there are multiple such letter permutations, you should choose the one where string (l_i, r_i) will be lexicographically minimum. If no such permutation exists, you should ignore the query (that is, not change string s).

Everybody knows that on TorCoder rounds input line and array size limits never exceed 60, so Leo solved this problem easily. Your task is to solve the problem on a little bit larger limits. Given string s and m queries, print the string that results after applying all m queries to string s.

Input

The first input line contains two integers n and m $(1 \le n, m \le 10^5)$ — the string length and the number of the queries.

The second line contains string s, consisting of n lowercase Latin letters.

Each of the next m lines contains a pair of integers l_i, r_i $(1 \le l_i \le r_i \le n)$ — a query to apply to the string.

Output

In a single line print the result of applying m queries to string s. Print the queries in the order in which they are given in the input.

Examples

input

```
7 2
aabcbaa
1 3
5 7
```

output

```
abacaba
```

input

```
3 2
abc
1 2
2 3
```

output

```
abc
```

Note

A *substring* (l_i, r_i) $1 \le l_i \le r_i \le n)$ of string $s = s_1 s_2 \dots s_n$ of length n is a sequence of characters $s_{l_i} s_{l_i + 1} \dots s_{r_i}$.

A string is a *palindrome*, if it reads the same from left to right and from right to left.

String $x_1 x_2 \dots x_p$ is *lexicographically smaller* than string $y_1 y_2 \dots y_q$, if either $p < q$ and $x_1 = y_1, x_2 = y_2, \dots, x_p = y_p$, or exists such number r $(r < p, r < q)$, that $x_1 = y_1, x_2 = y_2, \dots, x_r = y_r$ and $x_{r+1} < y_{r+1}$.