You are given a string S, which consists entirely of decimal digits (0–9). Using digits from S, create a palindromic number with the largest possible decimal value. You should use at least one digit and you can reorder the digits. A palindromic number remains the same when its digits are reversed; for instance, "7", "44" or "83238". Any palindromic number you create should not, however, have any leading zeros, such as in "0990" or "010".

For example, decimal palindromic numbers that can be created from "8199" are: "1", "8", "9", "99", "919" and "989". Among them, “989” has the largest value.

Write a function:

```
def solution(S)
```

that, given a string S of N digits, returns the string representing the palindromic number with the largest value.

Examples:

1. Given "39878", your function should return "898".

2. Given "00900", your function should return "9".

3. Given "0000", your function should return "0".

4. Given "54321", your function should return "5".

Write an efficient algorithm for the following assumptions:

- N is an integer within the range [1..100,000];
- string S is made only of digits (`0`–`9`).