String: Largest String

Given is a string ***S***, consisting of lower case distinct characters.
You have to create a lexicographically largest string that satisfies the condition:
string F = X + reverse(Y).
where "+" denotes concatenation and it satisfies the following conditions.

- length of string ***F* <= length of (S)**
- ***X*** is the prefix substring of string ***S*** and ***Y*** is the suffix substring of string ***S***.

Note
X and ***Y*** can also be empty.
Lexicographic order is the way of the ordering of words based on the alphabetical order of English letters, i.e. "a" is the smallest letter and "z" is the greatest letter.
Length of string ***S*** is at most 26.

Function Description
In the provided code snippet, implement the provided `largestString(...)` method using the variables to print the lexicographically largest string. You can write your code in the space below the phrase `“WRITE YOUR LOGIC HERE”`.

There will be multiple test cases running so the Input and Output should match exactly as provided.
The base Output variable `result` is set to a default value of `-404` which can be modified. Additionally, you can add or remove these output variables.

Input Format
The first line contains a string ***S***.

Sample Input

```
dacb    -- denotes string S
```

Constraints
1 <= length(***S***) <= 26

Output Format
The output contains the answer string denoting the lexicographically largest string ***F***.

Sample Output

```
dbca
```

Explanation
For given input string dacb,
Take ***X***(prefix string) = d
Take ***Y***(suffix string) = acb
Therefore, F = ***X*** + reverse(***Y***)

String: Largest String

Given is a string ***S***, consisting of lower case distinct characters.

You have to create a lexicographically largest string that satisfies the condition:
string F = X + reverse(Y).
where "+" denotes concatenation and it satisfies the following conditions.

- length of string ***F* <= length of (S)**
- ***X*** is the prefix substring of string ***S*** and ***Y*** is the suffix substring of string ***S***.

Note

X and ***Y*** can also be empty.

Lexicographic order is the way of the ordering of words based on the alphabetical order of English letters, i.e. "a" is the smallest letter and "z" is the greatest letter.

Length of string ***S*** is at most 26.

Function Description

In the provided code snippet, implement the provided `largestString(...)` method using the variables to print the lexicographically largest string. You can write your code in the space below the phrase `“WRITE YOUR LOGIC HERE”`.

There will be multiple test cases running so the Input and Output should match exactly as provided.
The base Output variable `result` is set to a default value of `-404` which can be modified. Additionally, you can add or remove these output variables.

Input Format

The first line contains a string ***S***.

Sample Input

```
dacb    -- denotes string S
```

Constraints

1 <= length(***S***) <= 26

Output Format

The output contains the answer string denoting the lexicographically largest string ***F***.

Sample Output

```
dbca
```

Explanation

For given input string dacb,
Take ***X***(prefix string) = d
Take ***Y***(suffix string) = acb
Therefore, F = ***X*** + reverse(***Y***)

String: Largest String

Given is a string ***S***, consisting of lower case distinct characters.
You have to create a lexicographically largest string that satisfies the condition:
string F = X + reverse(Y).
where "+" denotes concatenation and it satisfies the following conditions.

- length of string ***F*** **<= length of (S)**
- ***X*** is the prefix substring of string ***S*** and ***Y*** is the suffix substring of string ***S***.

Note
X and ***Y*** can also be empty.
Lexicographic order is the way of the ordering of words based on the alphabetical order of English letters, i.e. "a" is the smallest letter and "z" is the greatest letter.
Length of string ***S*** is at most 26.

Function Description
In the provided code snippet, implement the provided `largestString(...)` method using the variables to print the lexicographically largest string. You can write your code in the space below the phrase `“WRITE YOUR LOGIC HERE”`.

There will be multiple test cases running so the Input and Output should match exactly as provided.
The base Output variable `result` is set to a default value of `-404` which can be modified. Additionally, you can add or remove these output variables.

Input Format
The first line contains a string ***S***.

Sample Input

```
dacb    -- denotes string S
```

Constraints
1 <= length(***S***) <= 26

Output Format
The output contains the answer string denoting the lexicographically largest string ***F***.

Sample Output

```
dbca
```

Explanation
For given input string dacb,
Take ***X***(prefix string) = d
Take ***Y***(suffix string) = acb
Therefore, F = ***X*** + reverse(***Y***)

String: Largest String

Given is a string ***S***, consisting of lower case distinct characters.
You have to create a lexicographically largest string that satisfies the condition:
string F = X + reverse(Y).
where "+" denotes concatenation and it satisfies the following conditions.

- length of string ***F*** **<= length of (S)**
- ***X*** is the prefix substring of string ***S*** and ***Y*** is the suffix substring of string ***S***.

Note
X and ***Y*** can also be empty.
Lexicographic order is the way of the ordering of words based on the alphabetical order of English letters, i.e. "a" is the smallest letter and "z" is the greatest letter.
Length of string ***S*** is at most 26.

Function Description
In the provided code snippet, implement the provided `largestString(...)` method using the variables to print the lexicographically largest string. You can write your code in the space below the phrase `“WRITE YOUR LOGIC HERE”`.

There will be multiple test cases running so the Input and Output should match exactly as provided.
The base Output variable `result` is set to a default value of `-404` which can be modified. Additionally, you can add or remove these output variables.

Input Format
The first line contains a string ***S***.

Sample Input

```
dacb    -- denotes string S
```

Constraints
1 <= length(***S***) <= 26

Output Format
The output contains the answer string denoting the lexicographically largest string ***F***.

Sample Output

```
dbca
```

Explanation
For given input string dacb,
Take ***X***(prefix string) = d
Take ***Y***(suffix string) = acb
Therefore, F = ***X*** + reverse(***Y***)