**Problem Statement: Maximize Your Score**

You are given a string number which is your score on a test and a character digit. Your task is to print the resulting score after removing exactly one occurrence of digit from your score such that the value of the resulting score in decimal form is maximized. The test cases are generated such that the digit occurs at least once in number.

**Input Format**

* The first line contains a string number.
* The second line contains a character digit.

**Constraints**

* 1 <= ∣number∣ <= 10,000
* '1' <= digit <= '9'
* digit occurs at least once in number

**Output Format**

A string representing your maximum score.

**Sample Input**

14323938

3

**Sample Output**

1432938

**Explanation**

All possible values of the score after removing a single '3' from the original string 14323938 are:

* 1423938
* 1432938
* 1432398

The maximum score among these is 1432938.

**Solution**

The approach to solving this problem involves:

1. Iterating over the string and identifying positions where the digit occurs.
2. For each position, create a new string by removing the digit at that position.
3. Compare the resulting strings and return the maximum one.

Here's a Python implementation of the solution:

python

def maximize\_score(number: str, digit: str) -> str:

max\_score = ""

for i in range(len(number)):

if number[i] == digit:

new\_score = number[:i] + number[i+1:]

if new\_score > max\_score:

max\_score = new\_score

return max\_score

# Example usage

number = "14323938"

digit = "3"

print(maximize\_score(number, digit)) # Output: 1432938

**Detailed Explanation of the Solution:**

1. **Initialization**: Start with an empty string max\_score which will store the maximum score found.
2. **Iteration**: Loop through each character in the string number.
   * If the current character matches the digit, create a new string new\_score by removing the digit at the current position.
   * Compare new\_score with max\_score and update max\_score if new\_score is greater.
3. **Return Result**: After looping through the entire string, return max\_score.

This approach ensures that we check all possible removals of the given digit and find the one which maximizes the score.

**Additional Test Cases**

**Test Case 1**

**Input:**

233

3

**Output:**

23

**Explanation:** Possible values: 23, 23. The maximum score is 23.

**Test Case 2**

**Input:**

9999

9

**Output:**

999

**Explanation:** Possible values: 999, 999, 999, 999. The maximum score is 999.

**Test Case 3**

**Input:**

123456

2

**Output:**

13456

**Explanation:** Possible values: 13456. The maximum score is 13456

**Test Case 4**

**Input:**

12342562

2

**Output:**

1342562

**Test Case 5**

**Input:**

51234255625

5

**Output:**

5123425625

**Test Case 6**

**Input:**

5144556245

4

**Output:**

514556245