Solved Problem - Compare Numbers

In this lesson, we'll discuss another string manipulation problem.

We'll cover the following

Problem statement

Given two very long integers A and B. You have to determine which is bigger or if they are equal. *Leading zeroes are allowed.*

Input format

The first line contains a non-negative integer, A.

The second line contains a non-negative integer, B.

A,B may contain leading zeroes. Each of them consists of no more than 10^6 digits.

Output format

Print a single character:

- `A` - if $A > B$
- `B` - if $B > A$
- `=` - If numbers are equal

Sample

Input 1

```
6
19
```

Output 1

```
B
```

Input 2

```
23567
0023534
```

Output 2

```
A
```

Input 3

```
63
0063
```

Output 3

```
=
```

Explanation

Sample 2: The numbers without leading zeros are `23` and `12`. We'll print `A` since $23 > 12$.

Solution

First of all, the number of digits can be 10^6. So clearly, no integer data type is enough *(`int` is 10 digits)*. We **have** to process the input as a string.

Before comparing the string, we remove the trailing zeros from both the numbers.

Comparing two numbers as strings without leading zeros, the

Comparing two numbers as a string without leading zeros has three cases:

1. $len(A) > len(B)$ - A is larger
2. $len(A) < len(B)$ - B is larger
3. $len(A) = len(B)$ - First digit from the left that is different. If all digits are same, print `=`.

Removing leading zeros

Removing the first character one by one in order to remove the leading zeroes will be an $O(N^2)$ operation. A better way to do this would be to create a new string without leading zeros.

This would remove leading zeros in $O(N)$.

```
#include <iostream>
#include <string>
using namespace std;

string remove_leading_zeroes(string s) {
  string ret;
  bool leading_zero = true;
  for (char c : s) {
    if (c != '0')
      leading_zero = false;
    if (!leading_zero)
     ret += c;
  }
  return ret;
}

char compare_numbers(string A, string B) {
  string a = remove_leading_zeroes(A);
  string b = remove_leading_zeroes(B);

  if (a.size() != b.size())
    return a.size() > b.size() ? 'A' : 'B';

  for (int i = 0; i < a.size(); i++) {
    if (a[i] == b[i])
      continue;
    return a[i] > b[i] ? 'A' : 'B';
  }

  return '=';
}

int main() {

  cout << compare_numbers("6", "19") << "\n";
  cout << compare_numbers("23576", "0023534") << "\n";
  cout << compare_numbers("63", "0063") << "\n";
  return 0;
}
```

With this problem, we’ll end the chapter on strings. In the next chapter we’ll discuss sorting algorithms.