

Find next greater number with same set of digits

Given a number n , find the smallest number that has same set of digits as n and is greater than n . If x is the greatest possible number with its set of digits, then print "not possible".

Examples:

For simplicity of implementation, we have considered input number as a string.

Input: $n = "218765"$

Output: "251678"

Input: $n = "1234"$

Output: "1243"

Input: $n = "4321"$

Output: "Not Possible"

Input: $n = "534976"$

Output: "536479"

We strongly recommend to minimize the browser and try this yourself first.

Following are few observations about the next greater number.

- 1) If all digits sorted in descending order, then output is always "Not Possible". For example, 4321.
- 2) If all digits are sorted in ascending order, then we need to swap last two digits. For example, 1234.
- 3) For other cases, we need to process the number from rightmost side (why? because we need to find the smallest of all greater numbers)

You can now try developing an algorithm yourself.

Following is the algorithm for finding the next greater number.

I) Traverse the given number from rightmost digit, keep traversing till you find a digit which is smaller than the previously traversed digit. For example, if the input number is “534976”, we stop at **4** because 4 is smaller than next digit 9. If we do not find such a digit, then output is “Not Possible”.

II) Now search the right side of above found digit ‘d’ for the smallest digit greater than ‘d’. For “53**4**976”, the right side of 4 contains “976”. The smallest digit greater than 4 is **6**.

III) Swap the above found two digits, we get 53**6**97**4** in above example.

IV) Now sort all digits from position next to ‘d’ to the end of number. The number that we get after sorting is the output. For above example, we sort digits in bold 536**974**. We get “536**479**” which is the next greater number for input 534976.

Following is C++ implementation of above approach.

```
// C++ program to find the smallest number which greater
// and has same set of digits as given number
```

```
#include <iostream>
#include <cstring>
#include <algorithm>
using namespace std;
```

```
// Utility function to swap two digits
```

```
void swap(char *a, char *b)
{
    char temp = *a;
    *a = *b;
    *b = temp;
}
```

```
// Given a number as a char array number[], this function
// next greater number. It modifies the same array to s
```

```
void findNext(char number[], int n)
{
```

```
    int i, j;
```

```
    // I) Start from the right most digit and find the f
    // smaller than the digit next to it.
```

```
    for (i = n-1; i > 0; i--)
```

```

        if (number[i] > number[i-1])
            break;

// If no such digit is found, then all digits are in
// means there cannot be a greater number with same :
if (i==0)
{
    cout << "Next number is not possible";
    return;
}

// II) Find the smallest digit on right side of (i-1)
// greater than number[i-1]
int x = number[i-1], smallest = i;
for (j = i+1; j < n; j++)
    if (number[j] > x && number[j] < number[smallest])
        smallest = j;

// III) Swap the above found smallest digit with number[i-1]
swap(&number[smallest], &number[i-1]);

// IV) Sort the digits after (i-1) in ascending order
sort(number + i, number + n);

cout << "Next number with same set of digits is " <<
return;
}

```

// Driver program to test above function

```

int main()
{
    char digits[] = "534976";
    int n = strlen(digits);
    findNext(digits, n);
    return 0;
}

```

Output:

Next number with same set of digits is 536479

The above implementation can be optimized in following ways.

1) We can use binary search in step II instead of linear search.

2) In step IV, instead of doing simple sort, we can apply some clever technique to do it in linear time. Hint: We know that all digits are linearly sorted in reverse order except one digit which was swapped.