Lexicographic order in C++

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

void dfs(int i, int n) {
    if (i > n) {
        return;
    }
    cout << i << endl;
    for (int j = 0; j < 10; j++) {
        dfs(10 * i + j, n);
    }
}

int main() {
    int n = 40;
    for (int i = 1; i <= 9; i++) {
        dfs(i, n);
    }
    return 0;
}
```

Initial Setup:

We begin by calling dfs(i, 20) for i = 1 to i = 9.

Dry Run (for n = 20):

1. Calling dfs(1, 20):

- The function prints 1.
- Then it recursively calls dfs(10, 20), dfs(11, 20), ..., dfs(19, 20).

Step by step:

- dfs(1, 20):
    - Prints 1.
    - Calls dfs(10, 20):
        - Prints 10.
        - Calls dfs(100, 20), but 100 > 20, so this call ends.
    - Calls dfs(11, 20):
        - Prints 11.
        - Calls dfs(110, 20), but 110 > 20, so this call ends.
    - Calls dfs(12, 20):
        - Prints 12.
        - Calls dfs(120, 20), but 120 > 20, so this call ends.
    - Calls dfs(13, 20):
        - Prints 13.
        - Calls dfs(130, 20), but 130 > 20, so this call ends.
    - Calls dfs(14, 20):
        - Prints 14.
        - Calls dfs(140, 20), but 140 > 20, so this call ends.
    - Calls dfs(15, 20):
        - Prints 15.
        - Calls dfs(150, 20), but 150 > 20, so this call ends.
    - Calls dfs(16, 20):
        - Prints 16.
        - Calls dfs(160, 20), but 160 > 20, so this call ends.
    - Calls dfs(17, 20):
        - Prints 17.
        - Calls dfs(170, 20), but 170 > 20, so this call ends.
    - Calls dfs(18, 20):
        - Prints 18.
        - Calls dfs(180, 20), but 180 > 20, so this call ends.
    - Calls dfs(19, 20):
        - Prints 19.
        - Calls dfs(190, 20), but 190 > 20, so this call ends.

2. Calling dfs(2, 20):

- The function prints 2.
- Then it recursively calls dfs(20, 20).

Step by step:

- dfs(2, 20):
  - Prints 2.
  - Calls dfs(20, 20):
    - Prints 20.
    - Calls dfs(200, 20), but 200 > 20, so this call ends.

At this point, the function has printed the number starting with 2:

Output:-

```
1
10
11
12
13
14
15
16
17
18
19
2
20
3
4
5
6
7
8
9
```