Largest after k swaps in C++

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

string max_str;

void findMaximum(string str, int k) {
   // Base case: When k swaps are used up
   if (k == 0) {
      return;
   }

   int n = str.length();

   // Find the maximum digit available for current
position
   for (int i = 0; i < n - 1; i++) {
      for (int j = i + 1; j < n; j++) {
         // If digit at position j is greater than digit at
position i, swap them
         if (str[j] > str[i]) {
            swap(str[i], str[j]);

            // Check if current string is larger than
previously found max
            if (str > max_str) {
               max_str = str;
            }

            // Recur for k-1 swaps on the modified string
            findMaximum(str, k - 1);

            // Backtrack: Swap again to revert to
original string
            swap(str[i], str[j]);
         }
      }
   }
}

int main() {
   string str = "1234567";
   int k = 4;

   // Initialize max_str with the original string
   max_str = str;

   // Find the maximum number possible after k
swaps
   findMaximum(str, k);

   // Print the maximum number found
   cout << max_str << endl;

   return 0;
}
```

Dry Run of the Code

Input:

- str = "1234567"
- k = 4

Step-by-Step Execution

Initialization

- max_str = "1234567"

First Level (k = 4)

- Outer loop: i = 0
  - Inner loop: j = 1
    - Swap: "2134567"
      - max_str = "2134567"
      - Recur with k = 3.

Second Level (k = 3)

- Outer loop: i = 0
  - Inner loop: j = 1
    - No swap (digits are the same).
  - Inner loop: j = 2
    - Swap: "3124567"
      - max_str = "3124567"
      - Recur with k = 2.

Third Level (k = 2)

- Outer loop: i = 0
  - Inner loop: j = 1, j = 2: No change (smaller results).
  - Inner loop: j = 3
    - Swap: "4123567"
      - max_str = "4123567"
      - Recur with k = 1.

Fourth Level (k = 1)

	• Outer loop: i = 0
	  ○ Inner loop: j = 4
	  ▪ Swap: "5123467"
	  ▪ max_str = "5123467"
	  ▪ Recur with k = 0.
	
	Base Case (k = 0)
	
	• Stop recursion and backtrack.
	
	Backtracking
	
	• Undo each swap and check other combinations.
	• Repeat similar logic for other positions (i = 1, 2, ...).
	
	Final max_str
	
	The largest number found after 4 swaps is:
	7654321

Output:- 7654321	