|  |  |
| --- | --- |
| **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;  } | **Explanation of the Algorithm:**   * For every pair (i, j) where i < j, if str[j] > str[i], swap i and j. * After each swap, check if the new number is greater than the current max\_str. * Recurse with k - 1. * Backtrack (swap back) to explore other options.  | **Call#** | **k** | **Swap Made (i↔j)** | **str After Swap** | **max\_str Before** | **max\_str After** | **Remarks** | | --- | --- | --- | --- | --- | --- | --- | | 1 | 4 | 0↔6 | 7234561 | 1234567 | ✅ 7234561 | New max | | 2 | 3 | 1↔5 | 7634521 | 7234561 | ✅ 7634521 | New max | | 3 | 2 | 2↔4 | 7654321 | 7634521 | ✅ 7654321 | Final max | | 4 | 1 | No beneficial swap | - | 7654321 | 7654321 | Stop recursion | | 5 | 3 | 1↔4 | 7534261 | 7654321 | ❌ | Not greater | | 6 | 3 | 1↔3 | 7435261 | 7654321 | ❌ | Not greater | | 7 | 2 | 2↔3 (from 7435261) | 7453261 | 7654321 | ❌ | Still not better | | ... | - | ... | ... | ... | ... | Many paths explored |   We only continue recursion when beneficial. As you can see, once 7654321 is reached, **no further recursion produces a better result**, so that becomes the final output. 🏁 Final Output: 7654321 |
| Output:- 7654321 | |