70. Permutation Sequence

The set `[1, 2, 3, ..., n]` contains a total of `n!` unique permutations.

By listing and labeling all of the permutations in order, we get the following sequence for `n = 3`:

1. `"123"`
2. `"132"`
3. `"213"`
4. `"231"`
5. `"312"`
6. `"321"`

Given `n` and `k`, return the `kth` permutation sequence.

Example 1:

Input: n = 3, k = 3

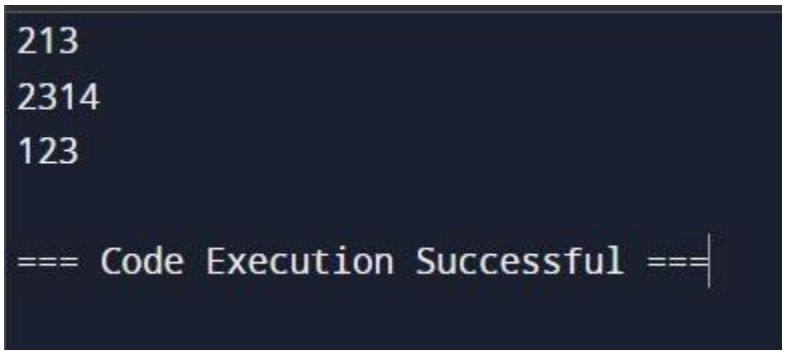
Output: "213"

PROGRAM:-

```
def getPermutation(n, k):
    import math

    # Generate the initial list of numbers
    numbers = list(range(1, n + 1))
    # Convert k to zero-based index
    k -= 1
    # Initialize the result
    result = []
    # Calculate the factorial values up to (n-1)!
    factorial = [1] * n
    for i in range(1, n):
        factorial[i] = factorial[i - 1] * i

    # Construct the kth permutation
    for i in range(n, 0, -1):
        # Determine the index of the current digit
```

```
        index = k // factorial[i - 1]
        # Append the digit to the result
        result.append(numbers[index])
        # Remove the used digit from the list
        numbers.pop(index)
        # Reduce k
        k %= factorial[i - 1]

    # Join the result list to form the final permutation string
    return ''.join(map(str, result))

# Example usage and output
n1, k1 = 3, 3
print(getPermutation(n1, k1))  # Output: "213"

n2, k2 = 4, 9
print(getPermutation(n2, k2))  # Output: "2314"

n3, k3 = 3, 1
print(getPermutation(n3, k3))  # Output: "123"
```

OUTPUT:-



TIME COMPLEXITY:-O(n^2)