

Program 72. **Permutation Sequence**

The set  $[1, 2, 3, \dots, n]$  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  $k$ th permutation sequence.

**Example 1:**

**Input:**  $n = 3, k = 3$

**Output:** "213"

Program:

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

    # Create a list of numbers to get permutations from
    nums = list(range(1, n + 1))

    # Convert k to be zero-based
    k -= 1

    # Variable to store our answer
    result = []

    # Loop through each position to determine the correct digit
    for i in range(n, 0, -1):
        # Find the factorial of the remaining digits
        fact = math.factorial(i - 1)

        # Determine the index of the next digit to use
        index = k // fact

        # Append that digit to our result
        result.append(str(nums[index]))

        # Remove that digit from the list
        nums.pop(index)

        # Adjust k to find the next digit
        k %= fact
```

```
return ".join(result)
```

# Example usage:

```
n = 3
```

```
k = 3
```

```
print(getPermutation(n, k)) # Output: "213"
```

Output:

```
"C:\Program Files\Python312\python.exe" "C:\Work Space\DAA\DAA COADS.PYTHON\program 72.py"
213
```

```
Process finished with exit code 0
```

Time complexity:

$O(n^2)$