Question 1:

Marks: 60

Problem Statement:

In the land of Wordsmithia, where words danced in the pages of enchanted dictionaries, a unique challenge arose. The wise sage Lexico faced a task involving the mystical art of lexicographical order.

Lexico explained, "Lexicographically means arranging things just like the words in a wizard's dictionary. For instance, consider two magical strings:

String 1: abcdef

String 2: abcdez

String 2 is lexicographically larger than String 1. The magic lies in the first position where they differ—'z' in String 2 comes after 'f' in String 1. We compare characters at this first differing position, and 'z' is greater than 'f' by their ASCII values. So, String 2 is lexicographically larger."

Now, imagine you have a string "s" and a magical number "k." Your quest is to find the lexicographically maximum possible string after removing k characters.

Let's embark on a couple of adventures:

Example 1:

Input: s=" fzlyoapx", K=2

Output: "zyoapx"

In this enchanting tale, the string "fzlyoapx" holds a secret. To make it as large as possible, we must remove two characters. Lexico, with a glint in his eye, identifies the humble 'f' and 'l' as the chosen

ones. By banishing 'f' and 'l' the string transforms into "zyoapx," standing tall as the lexicographically maximum.

Example 2:

Input: s = "xdycz", K=2

Output: "ycz"

In this mystical journey, we encounter the string "xdycz" and the challenge of removing two characters. To maximize its lexicographical greatness, we bid farewell to the first two characters, 'x' and 'd.' The residue, "ycz," emerges as the triumphant lexicographical zenith.

And remember, any other removals would lead to a smaller lexicographical string. Lexico **hints** at the use of a magical staff known as the **Stack**, ensuring a swift solution with optimal complexity **O(n + k).** Armed with this knowledge, adventurers set forth into the realm of strings, ready to maximize their lexicographical destiny. May the words guide you to victory!

Question 2:

Marks: 100

Problem Statement:

In the enchanted land of Numerica, where numbers orchestrated a magical symphony, a new quest unfurled before the adventurers. Digitara, the wise sorcerer, faced a challenge with the mystical number 'x.' The task was a unique operation—removing one digit at a time while adhering to specific rules.

The rules of this mystical operation were crystal clear:

- The resulting number must avoid any leading zeroes.
- The outcome must remain a positive integer.

Goal:

The adventurers aimed to unveil the smallest positive integer from 'x' by applying this operation 'k' times. The magical process unfolded as follows:

- Commence with the original number 'x.'
- Remove one digit, ensuring it follows the rules.
- Repeat this process 'k' times to reveal the smallest possible number.

An **example** illuminated the adventurers' path: **'x' as 10142, with 'k' set to 2.**

- Start with 10142, remove '4' to get 1012.
- Remove another digit, like '2,' to obtain 101.

- The result—the smallest positive integer achievable with 'k' operations—was 101. Notably, trailing zeroes were forbidden in this mystical arithmetic.

The quest continued with challenges represented by 'x' and 'k' values. The magical scrolls contained **sample tests**:

Example 1:

Input: x = 10000, k = 4

Output: 1

Example 2:

Input: x = 1337, k = 0

Output: 1337

Example 3:

Input: x = 987654321, k = 6

Output: 321

Example 4:

Input: x = 66837494128, k = 5

Output: 344128

Example 5:

Input: x = 7808652, k = 3

Output: 7052

In the mystical world of Numerica, where numbers loomed large, 'x' takes on the form of a string, much like a superhero donning a cape—it's not merely a cool choice; it's an essential step! (Using string for number x is necessary as the number x can be very very large).

Consider 'x' as a superhero number, transitioning into a string, akin to putting on a magical cape. **This shift grants flexibility to 'x,' which is crucial in Numerica**, where some numbers are so vast that our traditional methods of handling them in magic scrolls falter(some numbers are very large that they only can be handled by strings).

Thus, visualize 'x' as our superhero gearing up for an epic adventure—embracing a magical cape to navigate the grandest challenges with ease.

May your superhero strings save the day in the enchanted world of Numerica! A subtle **hint** echoed from Digitara's wisdom—**a Stack**, a magical tool with the power to weave solutions with optimal complexity (**O(n + k)**). The adventurers, well-acquainted from their exploits in Alphaburia, felt prepared for this new challenge. The journey into Numerica promised not only mathematical triumphs but also an exploration of the mystical dance between strings and numbers. Armed with their experiences, they ventured forth, poised to uncover the secrets of the smallest positive integers in the enchanted land of Numerica.

Note: Absolutely! The analogy is perfect. Solving the first problem in Alphaburia with Lexiconius laid the foundation for understanding the principles of lexicographical order and the strategic removal of characters.

The second problem in Numerica builds upon these concepts, introducing a new magical operation and a distinct set of challenges. The skills and approaches developed in Alphaburia serve as a valuable prerequisite, guiding adventurers through the enchanted realms of strings and numbers.

While the second problem may require some fresh thinking to navigate its unique intricacies, the familiarity with lexicographical order and the strategic removal of characters will undoubtedly prove handy. Think of it as advancing from one level

of magical expertise to the next—a progression where the knowledge gained in Alphaburia serves as a powerful wand in facing the challenges of Numerica.

With the wisdom acquired from Alphaburia and the new insights gained, adventurers are well-equipped to embark on this exciting journey, ready to unravel the mysteries of Numerica and claim victory over the enchanted integers.

Question 3:

Marks: 40

Problem Statement:

Imagine you are a financial analyst working for an investment firm, and your task is to identify the most lucrative investment opportunities in a series of daily stock prices. Each day, the stock prices are recorded, and you are given the flexibility to choose a sequence of 'k' consecutive days to maximize the profit.

Array of Stock Prices: Think of an array representing the daily stock prices, where each element signifies the closing price of the stock for a specific day.

[Day 1: $100, Day 2: $110, Day 3: $120, Day 4: $90, Day 5: $130, ...].

Integer 'k': In this financial analysis challenge, 'k' represents the number of consecutive days you can choose to buy and sell the stock. For instance, if 'k' is 3, you aim to find the sequence of three consecutive days that would result in the maximum profit.

Subsequence with the Highest Profit: The subsequence corresponds to selecting 'k' consecutive days with the highest cumulative difference in stock prices. The objective is to maximize the profit gained from buying and selling stocks during this period.

Queue Data Structure: Visualize a queue-like approach, where you keep track of the best 'k' consecutive days while moving through the array of stock prices. The algorithm dynamically adjusts the queue to include the most profitable 'k' days, ensuring you capitalize on the optimal trading opportunities.

Optimal Time Complexity: By implementing an algorithm inspired by the problem statement, you can efficiently analyze the stock prices and identify the 'k' consecutive days with the maximum profit. The optimal time complexity (**O(n + k) or O(n)**) ensures quick decision-making without unnecessary delays.

For example:

Example 1: If you have an **array [-1, -2, 6, 5, 4] and 'k' is 3,** the subsequence with the largest sum is [6, 5, 4], and you'd return that as the result,
meaning you need to return the array above so,
Output: [6, 5, 4]

Example 2: If you have an **array [-2,0,-1,0,-3] and 'k' is 3,** the subsequence with the largest sum is [0,-1,0], and you'd return that as the result,
meaning you need to return the array above so,
Output: [0,-,0]

In the fast-paced world of financial markets, this numeric quest becomes a powerful tool for analysts seeking to optimize investment strategies. By applying algorithms inspired by this challenge, analysts can make informed decisions, maximize profits, and contribute to the overall success of their investment portfolios.