Algorithms Laboratory (CS29203)
Assignment 4: Dynamic Programming
Department of CSE, IIT Kharagpur

5th September 2024

Question-1

(50 points)
Consider a string `S` consisting of only characters 'x' and 'y'. Consider a pair of indices (p, q). The string `S` is called *Kgp string* if there is no such pair of indices (p, q) so that $p < q$ and $S[p] = y$ and $S[q] = x$. Your task is to find the **minimum** number of deletion of characters needed from `S` to make it a *Kgp string*.

For example, consider the following string `S = "aababbab"`. Then the minimum number of deletions needed is 2 since you can delete characters at index 2 and 6 to make the string a *Kgp string* (assume 0-indexed convention). Note that there can be multiple possible deletions. In this case deletion of characters at indices 3 and 6 will also make the string a *Kgp string*.

Your task is to solve and implement using **dynamic programming** approach. *Hint: Think of keeping track of the minimum number of deletions needed so far as you traverse the string from left to right. There can be two possibilities: the current character is 'y', or it is 'x'. In the first case when the current character is 'y', no need to perform deletion currently; count the number of 'y'. Otherwise if the current character is 'x', you can either delete the current character or delete the previous 'y'. Hence you can choose the best of these two. In this way you can set the DP conditions.*

Example:

```
(Input) Enter the string: bbaaaaabb
(Output) Minimum number of deletion needed is 2
```

Question-2

(50 points)
Consider an integer array $\mathcal{S}[\,]$ having n elements in it. Let *w, x, y, z* are four indices of the array where $w > x > y > z$. Let the *Kgp value* of the array $\mathcal{K}$ is defined as

$$\mathcal{K} = \mathcal{S}[w] - \mathcal{S}[x] + \mathcal{S}[y] - \mathcal{S}[z].$$

Your task is to find the **maximum** possible *Kgp value* in the array $\mathcal{S}$. For example, let $\mathcal{S} = [3, 9, 10, 1, 30, 40]$. Then the maximum *Kgp value* is 46 because of the choice of the following indices: $w = 5, x = 3, y = 2, z = 0$ resulting in the value of $\mathcal{K} = 40 - 1 + 10 - 3 = 46$.

Your implementation **must** use the idea of **dynamic programming** to solve the problem having complexity not more than $O(n)$. (*Hint: think of creating 4 lookup tables storing the maximum values of s[w], s[w]-s[x], s[w]-s[x]+s[y], and s[w]-s[x]+s[y]-s[z]. Then figure out how to find the final answer from these tables.*)
Example:

```
(Input) Enter the number of elements: 10
        Enter the array S[ ]: 12  20  5  32  77  9  11  34  19  15

(Output) Maximum Kgp value is 97
```