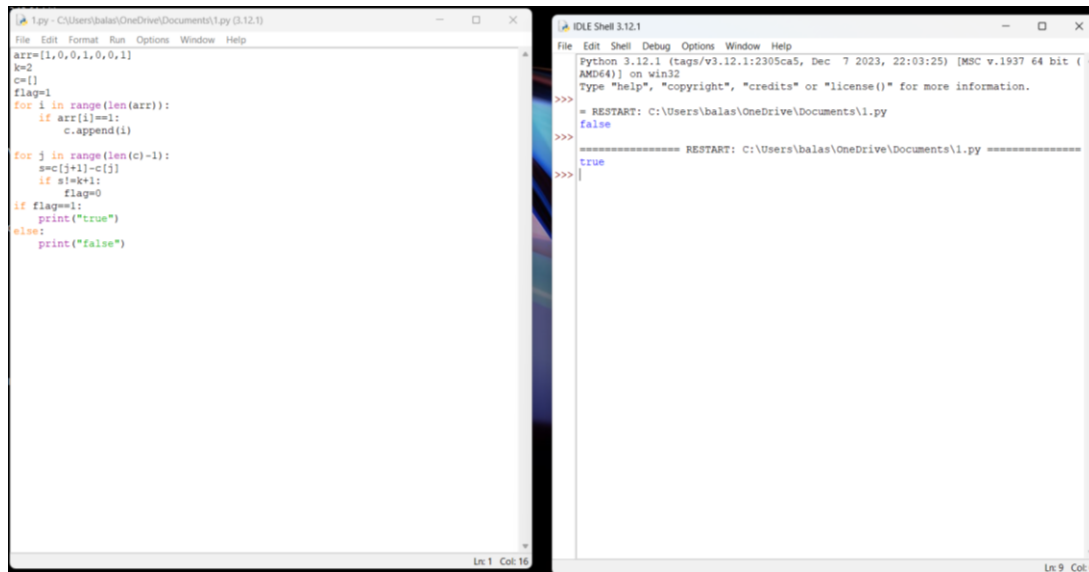


1. Given an binary array nums and an integer k, return true if all 1's are at least k places away from each other, otherwise return false.



```

1.py - C:\Users\balas\OneDrive\Documents\1.py (3.12.1)
File Edit Format Run Options Window Help
arr=[1,0,0,1,0,0,1]
k=2
c=[]
flag=1
for i in range(len(arr)):
    if arr[i]==1:
        c.append(i)

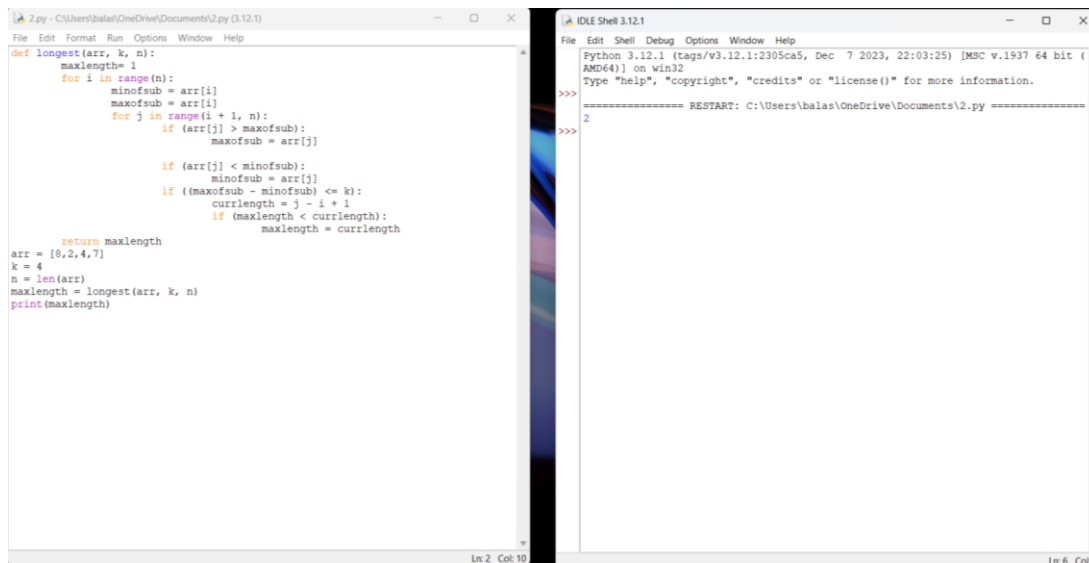
for j in range(len(c)-1):
    s=c[j+1]-c[j]
    if s<=k:
        flag=0
if flag==1:
    print("true")
else:
    print("false")
Ln: 1 Col: 16

IDLE Shell 3.12.1
File Edit Shell Debug Options Window Help
Python 3.12.1 (tags/v3.12.1:2305ca5, Dec 7 2023, 22:03:25) [MSC v.1937 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:\Users\balas\OneDrive\Documents\1.py =====
>>> false
>>>
===== RESTART: C:\Users\balas\OneDrive\Documents\1.py =====
>>> true
>>>
Ln: 9 Col: 0

```

2. Longest Continuous Subarray With Absolute Diff Less Than or Equal to Limit

Given an array of integers nums and an integer limit, return the size of the longest non-empty subarray such that the absolute difference between any two elements of this subarray is less than or equal to limit.



```

2.py - C:\Users\balas\OneDrive\Documents\2.py (3.12.1)
File Edit Format Run Options Window Help
def longest(arr, k, n):
    maxlength=1
    for i in range(n):
        minofsub = arr[i]
        maxofsub = arr[i]
        for j in range(i + 1, n):
            if (arr[j] > maxofsub):
                maxofsub = arr[j]

            if (arr[j] < minofsub):
                minofsub = arr[j]
            if ((maxofsub - minofsub) <= k):
                curlength = j - i + 1
                if (maxlength < curlength):
                    maxlength = curlength
        return maxlength
arr = [8,2,4,7]
k = 4
n = len(arr)
maxlength = longest(arr, k, n)
print(maxlength)
Ln: 2 Col: 10

IDLE Shell 3.12.1
File Edit Shell Debug Options Window Help
Python 3.12.1 (tags/v3.12.1:2305ca5, Dec 7 2023, 22:03:25) [MSC v.1937 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:\Users\balas\OneDrive\Documents\2.py =====
>>> 2
>>>
Ln: 6 Col: 0

```

3. Find the Kth Smallest Sum of a Matrix With Sorted Rows

You are given an m x n matrix mat that has its rows sorted in non-decreasing order and an integer k.

You are allowed to choose exactly one element from each row to form an array.

Return the kth smallest array sum among all possible arrays.

The screenshot shows a Python IDE with two windows. The left window, titled '3.py - C:\Users\balas\OneDrive\Documents\3.py (3.12.1)', contains the following code:

```
mat = [[1, 3, 11], [2, 4, 6]]
k = 5

arr = [0]
for i in mat:
    arr = sorted(a + b for a in arr for b in i[:k][:k])

output = arr[-1]
print(output)
```

The right window, titled 'IDLE Shell 3.12.1', shows the execution output:

```
>>> Python 3.12.1 (tags/v3.12.1:2305ca5, Dec 7 2023, 22:03:25) [MSC v.1937 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> = RESTART: C:\Users\balas\OneDrive\Documents\3.py
>>> 7
```

4. Count Triplets That Can Form Two Arrays of Equal XOR

Given an array of integers arr.

We want to select three indices i, j and k where $(0 \leq i < j \leq k < \text{arr.length})$.

Let's define a and b as follows:

- $a = \text{arr}[i] \oplus \text{arr}[i + 1] \oplus \dots \oplus \text{arr}[j - 1]$
- $b = \text{arr}[j] \oplus \text{arr}[j + 1] \oplus \dots \oplus \text{arr}[k]$

Note that \oplus denotes the bitwise-xor operation.

Return the number of triplets (i, j and k) Where $a == b$.

The screenshot shows a Python IDE with two windows. The left window, titled '4.py - C:\Users\balas\OneDrive\Documents\4.py (3.12.1)', contains the following code:

```
def triplets(arr):
    n = len(arr)
    c = 0
    for i in range(n):
        xor = arr[i]
        for j in range(i + 1, n):
            xor ^= arr[j]
            if xor == 0:
                c += j - i
    return c

arr = [2, 3, 1, 6, 7]
print(triplets(arr))
```

The right window, titled 'IDLE Shell 3.12.1', shows the execution output:

```
>>> Python 3.12.1 (tags/v3.12.1:2305ca5, Dec 7 2023, 22:03:25) [MSC v.1937 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> = RESTART: C:\Users\balas\OneDrive\Documents\4.py
>>> 4
```

5. Minimum Time to Collect All Apples in a Tree

Given an undirected tree consisting of n vertices numbered from 0 to n-1, which has some apples in their vertices. You spend 1 second to walk over one edge of the tree. Return the minimum time in seconds you have to spend to collect all apples in the tree, starting at vertex 0 and coming back to this vertex.

The edges of the undirected tree are given in the array edges, where edges[i] = [ai, bi] means that exists an edge connecting the vertices ai and bi. Additionally, there is a boolean array hasApple, where hasApple[i] = true means that vertex i has an apple; otherwise, it does not have any apple.

```

5.py - C:\Users\balas\OneDrive\Documents\5.py (3.12.1)
File Edit Format Run Options Window Help
def mintime(n, edges, hasApple):
    from collections import defaultdict
    tree = defaultdict(list)
    for u, v in edges:
        tree[u].append(v)
        tree[v].append(u)

    def dfs(node, parent):
        total_time = 0
        for neighbor in tree[node]:
            if neighbor == parent:
                continue
            time = dfs(neighbor, node)
            if time > 0 or hasApple[neighbor]:
                total_time += time + 2
        return total_time

    return dfs(0, -1)

n = 7
edges = [[0,1],[0,2],[1,4],[1,5],[2,3],[2,6]]
hasApple = [False, False, True, False, True, True, False]
print(mintime(n, edges, hasApple))

IDLE Shell 3.12.1
Python 3.12.1 (tags/v3.12.1:2305ca5, Dec 7 2023, 22:03:25) [MSC v.1937 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:\Users\balas\OneDrive\Documents\5.py
>>>
8
>>>

```

6. Number of Ways of Cutting a Pizza

Given a rectangular pizza represented as a rows x cols matrix containing the following characters: 'A' (an apple) and '.' (empty cell) and given the integer k. You have to cut the pizza into k pieces using k-1 cuts.

For each cut you choose the direction: vertical or horizontal, then you choose a cut position at the cell boundary and cut the pizza into two pieces. If you cut the pizza vertically, give the left part of the pizza to a person. If you cut the pizza horizontally, give the upper part of the pizza to a person. Give the last piece of pizza to the last person.

Return the number of ways of cutting the pizza such that each piece contains at least one apple. Since the answer can be a huge number, return this modulo $10^9 + 7$.

```

6.py - C:\Users\balas\OneDrive\Documents\6.py (3.12.1)
File Edit Format Run Options Window Help
od = 10**9 + 7

def ways(pizza, k):
    rows, cols = len(pizza), len(pizza[0])
    pre = [[0] * (cols + 1) for _ in range(rows + 1)]
    for i in range(rows - 1, -1, -1):
        for j in range(cols - 1, -1, -1):
            pre[i][j] = (1 if pizza[i][j] == 'A' else 0) + pre[i+1][j] + pre[i][j+1] - pre[i+1][j+1]
    dp = [[0] * (k + 1) for _ in range(cols) for _ in range(rows)]
    for i in range(rows):
        for j in range(cols):
            if pre[i][j] > 0:
                dp[i][j][1] = 1

    for c in range(2, k + 1):
        for i in range(rows):
            for j in range(cols):
                for x in range(i + 1, rows):
                    if pre[i][j] - pre[x][j] > 0:
                        dp[i][j][c] = (dp[i][j][c] + dp[x][j][c - 1]) % mod
                for y in range(j + 1, cols):
                    if pre[i][j] - pre[i][y] > 0:
                        dp[i][j][c] = (dp[i][j][c] + dp[i][y][c - 1]) % mod

    return dp[0][0][k]

pizza = ["A..", "AAA", "..."]
k = 3
print(ways(pizza, k))

IDLE Shell 3.12.1
Python 3.12.1 (tags/v3.12.1:2305ca5, Dec 7 2023, 22:03:25) [MSC v.1937 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:\Users\balas\OneDrive\Documents\6.py
>>>
3
>>>

```