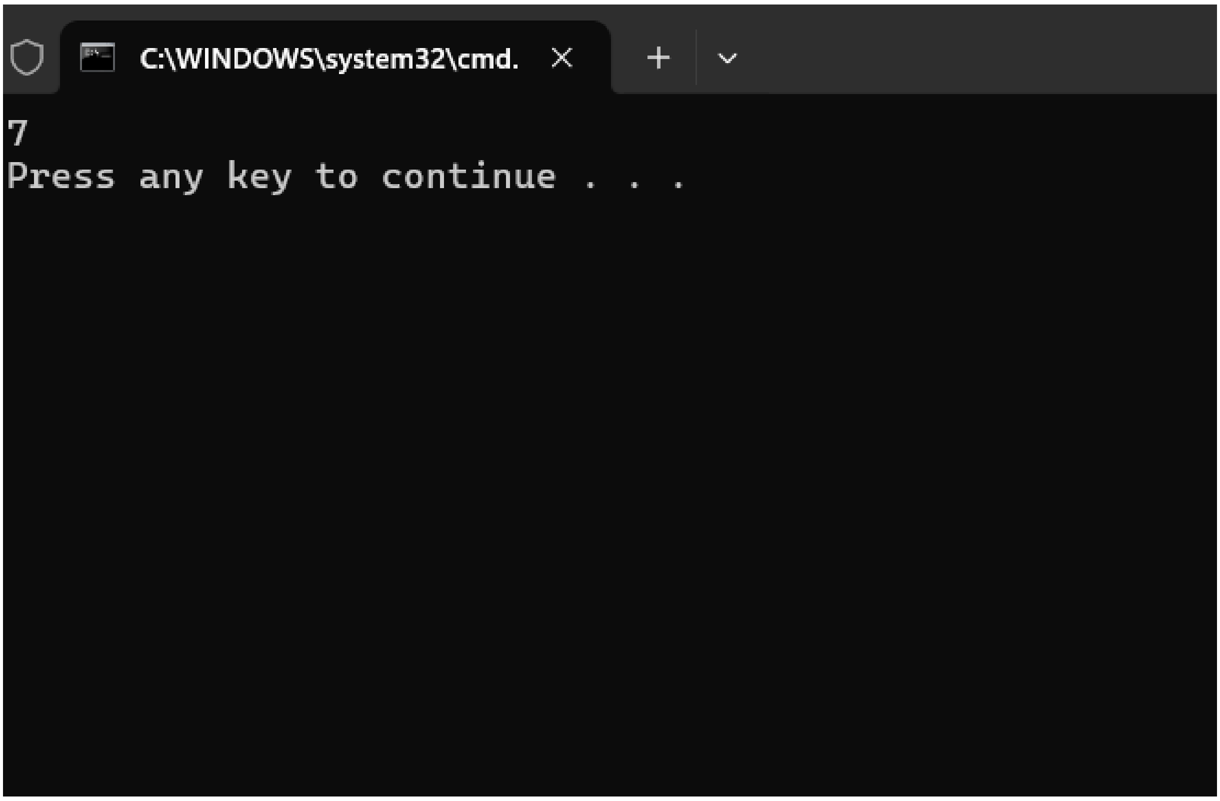
57) . Find the Kth Smallest Sum of a Matrix With Sorted Rows
You are given an m x n matrix mat that has its rows sorted in non-decreasing order and an integer k.
You are allowed to choose exactly one element from each row to form an array.
Return the kth smallest array sum among all possible arrays.

CODE:

```
import heapq

def kthSmallest(mat, k):
    m, n = len(mat), len(mat[0])
    min_heap = [(sum(row[0] for row in mat), [0] * m)]
    visited = set(tuple([0] * m))

    for _ in range(k):
        curr_sum, indices = heapq.heappop(min_heap)

        for i in range(m):
            if indices[i] + 1 < n:
                new_indices = list(indices)
                new_indices[i] += 1
                new_sum = curr_sum - mat[i][indices[i]] + mat[i][new_indices[i]]
                new_tuple = tuple(new_indices)
                if new_tuple not in visited:
                    visited.add(new_tuple)
                    heapq.heappush(min_heap, (new_sum, new_indices))

    return curr_sum

mat = [[1,3,11],[2,4,6]]
k = 5
print(kthSmallest(mat, k))
```

OUTPUT:



```
7
Press any key to continue . . .
```

TIME COMPLEXITY : **O(klogk)**