```
'''
Optimizing Reservoir Releases by Minimizing the Maximum release from a  reservoir

Problem:
    Efficiently operating reservoirs involves making decisions about water release,
storage, and inflow management. During flood seasons, reservoirs must balance
the need for dam safety (releasing water) with minimizing downstream flood
risk (releasing as little water as possible).

Things to note:
- Choose the best optimal release
Approach:
1.) Create a function that determines the release policy to maximize the total inflow
2.) It should be display the optimal release
'''


def find_optimal_release(inflow, weights, k):
    n = len(inflow)
    dp = [[0] * (k + 1) for _ in range(n + 1)]

    # Initialize the DP table
    for i in range(1, n + 1):
        for j in range(1, k + 1):
            dp[i][j] = max(dp[i - 1][j], dp[i - 1][j - weights[i - 1]] + inflow[i - 1])

    # Backtrack to find the optimal release trajectory
    solution = []
    i = n
    j = k
    while j > 0 and i > 0:
        if dp[i][j] != dp[i - 1][j]:
            solution.append(i)
            j -= weights[i - 1]
        i -= 1
    return solution

# Test :
inflow = [10, 15, 20, 25, 30, 35, 40]  # Example inflow data (in cubic meter)
weights = [2, 3, 4, 5, 6, 7, 8]      # Example weights (release amounts in cubic meter)
k = 15                        # Maximum allowable release

optimal_release = find_optimal_release(inflow, weights, k)
print("Optimal release indices:", optimal_release)
```

```
Optimal release indices: [7, 5, 4]
```

This algorithm optimally determines the release policy to maximize the total inflow while ensuring that the total release does not exceed the maximum allowable release at any time step.