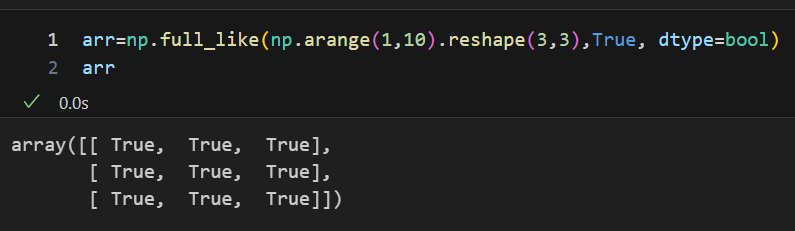
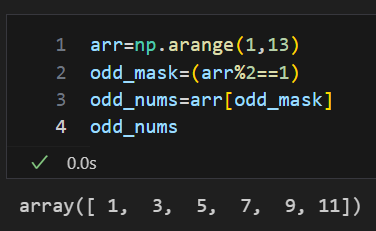
Assignment 6

Set 1:

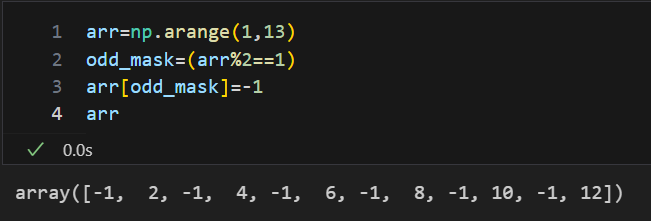
Create a 3×3 array of all True Boolean values



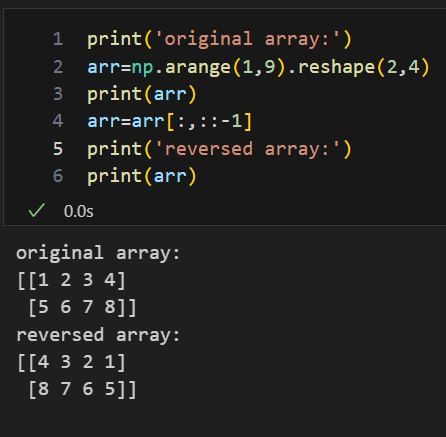
Extract odd numbers from an array



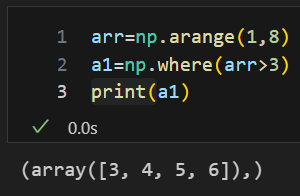
Replace all odd numbers with -1



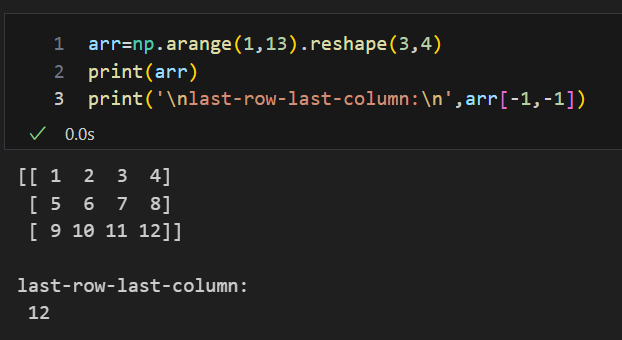
Reverse columns of a 2D array



Get indices of elements greater than 5 [hint: check where method of numpy]



Access last row - last column



Set 2:

Normalize a NumPy array [ to normalize means scaling the values of an array into a specific range, often [0, 1]]

Create a 5×5 array with 1 on border and 0 inside

Create a random array of 10×10: perform stats (min, max, mean, std)

Broadcast a 1D array over a 2D array [all possible scenarios]

Sort a 2D array by column 1 [hint: check method argsort()]

Set 3:

Find common elements between two arrays [with and without using intersect1d]

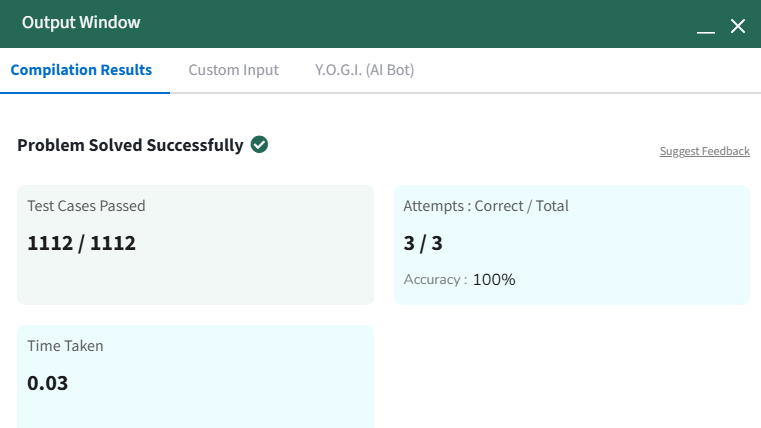
Subtract row mean from each row of a 2D array

Given a 4×4 array, swap the first half rows with the second half rows

Given a 1D array of random numbers, find the indices of the 5 largest values (no sorting/ check argsort() instead).

From a 2D NumPy array, extract only the unique rows (no duplicates hint: check unique method).

[Power of 2](https://www.geeksforgeeks.org/problems/power-of-2-1587115620/1)



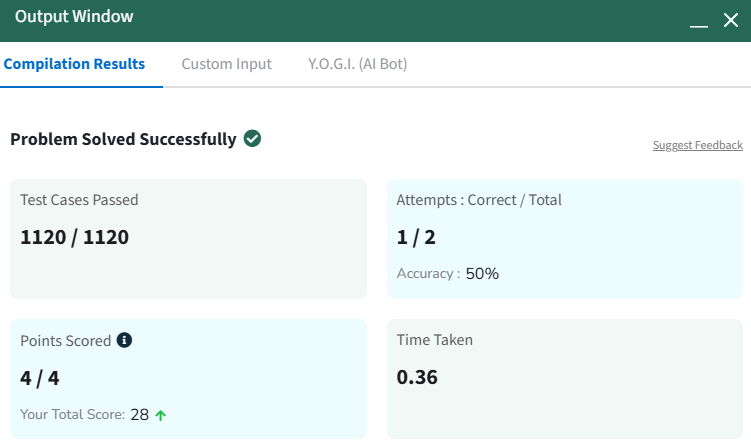
def is\_power\_of\_two\_bitwise(n):

    """Checks if n is a power of 2 using the bitwise AND operation."""

    # Must be positive (n > 0) AND have exactly one bit set to 1

    return n > 0 and (n & (n - 1) == 0)

[First and Last Occurrence](https://www.geeksforgeeks.org/problems/first-and-last-occurrences-of-x3116/1)



def find(arr, x):

        first\_o,last\_o=-1,-1

        if x not in set(arr):

            return [-1,-1]

        for i in range(len(arr)):

            if arr[i]==x & first\_o !=-1:

                print(i)

                first\_o,last\_o=i,i

            elif arr[i]==x:

                last\_o=i

        return [first\_o,last\_o]