**Midterm Exam Problems (1131)**

**Problem 1:**

Given an array of integers, write a function that shifts all elements of the array to the right by k positions. Elements that are shifted past the end of the array should appear at the beginning. Do this in-place without using additional arrays.

**Input:** An array of integers, arr[], and an integer k.

**Output:** The array shifted by ***k*** positions to the right.

**Example:**

Input: arr = {1, 2, 3, 4, 5}, **k** = 2 (***k*** is a positive integer.)

Output: arr = {4, 5, 1, 2, 3}

**Constraints:** The array length is between 1 and 100.

**Problem 2:**

Write a function that compresses a given string by replacing consecutive identical characters with the character followed by the count of repetitions. If the "compressed" string is not shorter than the original string, return the original.

**Input:** A string s (containing uppercase and lowercase letters only).

**Output:** A compressed string if it's shorter; otherwise, the original string.

**Example:**

**Input**: s = "aabcccccaaa"

**Output**: "a2b1c5a3"

**Input:** s = "abcd"

**Output:** "abcd" // Compression does not make the string shorter

**Constraints**: The length of s is between 1 and 100.

**Problem 3:**

Given an array of integers, find all unique pairs of elements that add up to a target sum. Print each pair only once, in ascending order, even if the array contains duplicates.

**Input:** An array of integers arr[] and an integer target.

**Output:** All unique pairs [x, y] where x + y = target, printed in ascending order (e.g., [3, 5] instead of [5, 3]).

**Example:**

**Input:** arr = {1, 3, 2, 2, 3, 5}, target = 6

**Output:** [1, 5], [3, 3]

**Constraints:** Array length between 1 and 1000.

Each element can be between -100 and 100.

**Problem 4:**

Given a square matrix, calculate the absolute difference between the sums of its diagonals.

**Input:** A 2D array matrix of size n x n.

**Output**: An integer representing the absolute difference.

**Example:**

**Input:**

11 2 4

4 5 6

10 8 -12

**Output**: 15 // Absolute difference = |11 + 5 - 12 - (4 + 5 + 10)| = 15

**Constraints:**

Matrix size n is between 2 and 1000.

**Problem 5:**

Given an array of non-negative integers and a target sum, determine if there is a contiguous subarray (of any length) that sums up to the target.

**Input:** An array of non-negative integers arr[] and an integer target.

**Output:** Return true if a subarray with sum equals target exists, otherwise return false.

**Examples:**

**Input**: arr = {1, 4, 20, 3, 10, 5}, target = 33

**Output**: true // because the subarray {20, 3, 10} sums to 33

**Input**: arr = {1, 4, 0, 0, 3, 10, 5}, target = 7

**Output**: true // subarray {4, 0, 0, 3} sums to 7

**Constraints:** Array length is between 1 and 100.

Each element can be between 0 and 100.