1. Palindrome Array Transformation

Write a program that reads 15 integers into an array. Transform the array to make it palindromic by replacing the second half with the mirror of the first half (middle element stays if odd size). Then allow the user to search for any number and report all positions where it appears.

Example: Input → `1 2 3 4 5 6 7 8 9 10 11 12 13 14 15` Output → `1 2 3 4 5 6 7 8 7 6 5 4 3 2 1`

---

2. Prime Number Filter and Statistics

Write a program that accepts 20 integers. Create two separate arrays: one storing only prime numbers from the input and another storing composite numbers. Display both arrays along with their averages. If a number appears multiple times, store it each time. Handle the special cases of 0, 1, and negative numbers appropriately.

---

3. Rotational Cipher Encryption

Develop a program that reads a sentence using `scanf("%[^\n]", str);` and encrypts it using a rotational cipher. Each letter should be shifted by a user-specified amount (1-25). Preserve spaces and punctuation. After encryption, allow the user to decrypt the message by reversing the shift. Do not use any string library functions.

Example: Shift = 3 Input → `Hello World!` Output → `Khoor Zruog!`

---

4. Array Partitioning Challenge

Write a program that reads 12 integers and partitions them into three arrays: one containing numbers divisible by both 2 and 3, one containing numbers divisible by either 2 or 3 (but not both), and one containing numbers divisible by neither. Display all three arrays with their element counts and calculate the sum of each partition.

---

5. Wave Pattern Sorting

Create a program that takes 10 integers and rearranges them in a "wave" pattern where `arr[0] >= arr[1] <= arr[2] >= arr[3] <= arr[4]...` without fully sorting the array. Use only comparisons and swaps. Display the original array, the wave-arranged array, and count how many swaps were needed.

Example: Input → `10 5 6 3 2 20 30 1 8 9` Possible Output → `10 5 6 2 20 3 30 1 9 8`

---

6. Character Frequency Analysis with Encoding

Write a program that reads a string using `scanf("%[^\n]", str);` and creates a frequency-encoded version. For each unique character, display the character followed by its count. Then provide a decoding mechanism where the user can enter the encoded string and get back the original format. Ignore spaces in counting.

Example: Input → `hello world` Encoded Output → `h1e1l3o2w1r1d1`

---

7. Subset Sum Finder

Write a program that reads 8 integers into an array. Then ask the user for a target sum. Find and display all possible pairs of elements whose sum equals the target. Count and display the total number of such pairs. Each element can be used only once per pair.

Example: Array → `2 4 3 5 7 8 9 1`, Target → `9` Output → `(2,7), (4,5), (8,1)` → 3 pairs found

---

8. Zigzag Merge Pattern

Create a program that reads two separate arrays of 6 integers each from the user. Merge them into a single array of 12 elements in a zigzag fashion: take first element from array1, first from array2, second from array1, second from array2, and so on. Then find and display the median value of the merged array.

Example: Array1 → `1 3 5 7 9 11` Array2 → `2 4 6 8 10 12` Merged → `1 2 3 4 5 6 7 8 9 10 11 12`

---

9. Pattern-Based String Validator

Write a program that reads a password using `scanf("%[^\n]", str);` and validates it based on these rules: must contain at least one uppercase letter, one lowercase letter, one digit, and one special character (!@#$%^&*). The password must be 8-15 characters long. Display which criteria are met and which are not. Do not use string library functions.

---

10. Sliding Window Maximum

Develop a program that reads 12 integers into an array. Ask the user for a window size (3-6). Then calculate and display the maximum value in every possible window of that size sliding through the array. Also display which window contains the overall maximum value.

Example: Array → `1 3 2 5 8 4 7 9 6 2 1 4`, Window = 3 Output → `3 5 8 8 8 9 9 9 6 4` (maximums of each window) Overall max window: `[7 9 6]` with max value `9`