**Array Program**

1.  **Write a Pseudo code to read and print elements of array.**

START

// Define the array and its size

DEFINE array[] // The array can be of any predefined size

DEFINE size // The number of elements in the array

// Read the number of elements

PRINT "Enter the number of elements:"

READ size

// Loop to input elements into the array

FOR i FROM 0 TO size - 1

PRINT "Enter element ", i + 1, ":"

READ array[i]

END FOR

// Loop to print the elements of the array

PRINT "The elements of the array are:"

FOR i FROM 0 TO size - 1

PRINT array[i]

END FOR

END

2. **Write a Pseudo code to print all negative elements in an array.**

START

// Define the array and its size

DEFINE array[] // The array can be of any predefined size

DEFINE size // The number of elements in the array

// Read the number of elements

PRINT "Enter the number of elements:"

READ size

// Loop to input elements into the array

FOR i FROM 0 TO size - 1

PRINT "Enter element ", i + 1, ":"

READ array[i]

END FOR

// Print negative elements

PRINT "Negative elements in the array are:"

FOR i FROM 0 TO size - 1

IF array[i] < 0 THEN

PRINT array[i]

END IF

END FOR

END

3.  **Write a Pseudo code to find sum of all array elements.**

START

// Define the array and its size

DEFINE array[] // The array can be of any predefined size

DEFINE size // The number of elements in the array

DEFINE sum // Variable to store the sum of the array elements

// Initialize sum to 0

sum = 0

// Read the number of elements

PRINT "Enter the number of elements:"

READ size

// Loop to input elements into the array

FOR i FROM 0 TO size - 1

PRINT "Enter element ", i + 1, ":"

READ array[i]

END FOR

// Calculate the sum of the elements

FOR i FROM 0 TO size - 1

sum = sum + array[i]

END FOR

// Print the sum

PRINT "The sum of all elements in the array is:", sum

END

4.  **Write a Pseudo code to find maximum and minimum element in an array.**

START

// Define the array and its size

DEFINE array[] // The array can be of any predefined size

DEFINE size // The number of elements in the array

DEFINE max // Variable to store the maximum element

DEFINE min // Variable to store the minimum element

// Read the number of elements

PRINT "Enter the number of elements:"

READ size

// Handle case where there are no elements

IF size <= 0 THEN

PRINT "Array is empty. Cannot determine maximum and minimum."

EXIT

END IF

// Read elements into the array

FOR i FROM 0 TO size - 1

PRINT "Enter element ", i + 1, ":"

READ array[i]

END FOR

// Initialize max and min with the first element of the array

max = array[0]

min = array[0]

// Loop to find the maximum and minimum elements

FOR i FROM 1 TO size - 1

IF array[i] > max THEN

max = array[i]

END IF

IF array[i] < min THEN

min = array[i]

END IF

END FOR

// Print the maximum and minimum elements

PRINT "The maximum element in the array is:", max

PRINT "The minimum element in the array is:", min

END

5.  **Write a Pseudo code to find second largest element in an array.**

START

// Define the array and its size

DEFINE array[] // The array can be of any predefined size

DEFINE size // The number of elements in the array

DEFINE max // Variable to store the maximum element

DEFINE second\_max // Variable to store the second maximum element

// Read the number of elements

PRINT "Enter the number of elements:"

READ size

// Handle case where there are not enough elements

IF size < 2 THEN

PRINT "Array must have at least two elements to find the second largest."

EXIT

END IF

// Read elements into the array

FOR i FROM 0 TO size - 1

PRINT "Enter element ", i + 1, ":"

READ array[i]

END FOR

// Initialize max and second\_max

max = -INFINITY

second\_max = -INFINITY

// Loop to find the maximum and second maximum elements

FOR i FROM 0 TO size - 1

IF array[i] > max THEN

second\_max = max

max = array[i]

ELSE IF array[i] > second\_max AND array[i] < max THEN

second\_max = array[i]

END IF

END FOR

// Check if second\_max has been updated

IF second\_max = -INFINITY THEN

PRINT "There is no second largest element (all elements might be the same)."

ELSE

// Print the second largest element

PRINT "The second largest element in the array is:", second\_max

END IF

END

6. **Write a Pseudo code to count total number of even and odd elements in an array.**

START

// Define the array and its size

DEFINE array[] // The array can be of any predefined size

DEFINE size // The number of elements in the array

DEFINE even\_count // Variable to count even elements

DEFINE odd\_count // Variable to count odd elements

// Initialize counters

even\_count = 0

odd\_count = 0

// Read the number of elements

PRINT "Enter the number of elements:"

READ size

// Read elements into the array

FOR i FROM 0 TO size - 1

PRINT "Enter element ", i + 1, ":"

READ array[i]

END FOR

// Loop to count even and odd elements

FOR i FROM 0 TO size - 1

IF array[i] MOD 2 = 0 THEN

even\_count = even\_count + 1

ELSE

odd\_count = odd\_count + 1

END IF

END FOR

// Print the counts

PRINT "Total number of even elements:", even\_count

PRINT "Total number of odd elements:", odd\_count

END

7.  **Write a Pseudo code to count total number of negative elements in an array.**

START

// Define the array and its size

DEFINE array[] // The array can be of any predefined size

DEFINE size // The number of elements in the array

DEFINE negative\_count // Variable to count negative elements

// Initialize the negative count

negative\_count = 0

// Read the number of elements

PRINT "Enter the number of elements:"

READ size

// Read elements into the array

FOR i FROM 0 TO size - 1

PRINT "Enter element ", i + 1, ":"

READ array[i]

END FOR

// Loop to count negative elements

FOR i FROM 0 TO size - 1

IF array[i] < 0 THEN

negative\_count = negative\_count + 1

END IF

END FOR

// Print the count of negative elements

PRINT "Total number of negative elements:", negative\_count

END

8.  **Write a Pseudo code to copy all elements from an array to another array.**

START

// Define the source and destination arrays and their size

DEFINE source\_array[] // The source array with elements

DEFINE destination\_array[] // The destination array to copy elements to

DEFINE size // The number of elements in the arrays

// Read the number of elements

PRINT "Enter the number of elements:"

READ size

// Initialize the destination array to the same size as the source array

DEFINE destination\_array[size]

// Read elements into the source array

FOR i FROM 0 TO size - 1

PRINT "Enter element ", i + 1, ":"

READ source\_array[i]

END FOR

// Copy elements from source\_array to destination\_array

FOR i FROM 0 TO size - 1

destination\_array[i] = source\_array[i]

END FOR

// Print the elements of the destination array

PRINT "Elements in the destination array are:"

FOR i FROM 0 TO size - 1

PRINT destination\_array[i]

END FOR

END

9. **Write a Pseudo code to insert an element in an array.**

START

// Define the array, its size, and the new element

DEFINE array[] // The array with elements

DEFINE size // The number of elements currently in the array

DEFINE new\_element // The new element to be inserted

DEFINE position // The position where the new element will be inserted

DEFINE i // Loop variable

// Read the current number of elements

PRINT "Enter the current number of elements:"

READ size

// Initialize the array to the given size

DEFINE array[size]

// Read elements into the array

FOR i FROM 0 TO size - 1

PRINT "Enter element ", i + 1, ":"

READ array[i]

END FOR

// Read the new element to be inserted

PRINT "Enter the element to be inserted:"

READ new\_element

// Read the position where the new element will be inserted

PRINT "Enter the position (0 to ", size, ") where the element should be inserted:"

READ position

// Validate the position

IF position < 0 OR position > size THEN

PRINT "Invalid position. The position should be between 0 and ", size

EXIT

END IF

// Shift elements to the right to make space for the new element

FOR i FROM size - 1 DOWNTO position

array[i + 1] = array[i]

END FOR

// Insert the new element

array[position] = new\_element

// Increment the size

size = size + 1

// Print the updated array

PRINT "The updated array is:"

FOR i FROM 0 TO size - 1

PRINT array[i]

END FOR

END

10. **Write a Pseudo code to delete an element from an array at specified position.**

START

// Define the array, its size, and the position of the element to delete

DEFINE array[] // The array with elements

DEFINE size // The number of elements currently in the array

DEFINE position // The position of the element to be deleted

DEFINE i // Loop variable

// Read the current number of elements

PRINT "Enter the current number of elements:"

READ size

// Initialize the array to the given size

DEFINE array[size]

// Read elements into the array

FOR i FROM 0 TO size - 1

PRINT "Enter element ", i + 1, ":"

READ array[i]

END FOR

// Read the position of the element to be deleted

PRINT "Enter the position (0 to ", size - 1, ") of the element to be deleted:"

READ position

// Validate the position

IF position < 0 OR position >= size THEN

PRINT "Invalid position. The position should be between 0 and ", size - 1

EXIT

END IF

// Shift elements to the left to fill the gap created by the deleted element

FOR i FROM position TO size - 2

array[i] = array[i + 1]

END FOR

// Decrease the size of the array

size = size - 1

// Print the updated array

PRINT "The updated array is:"

FOR i FROM 0 TO size - 1

PRINT array[i]

END FOR

END

11.  **Write a Pseudo code to count frequency of each element in an array.**

START

// Define the array, its size, and auxiliary structures

DEFINE array[] // The array with elements

DEFINE size // The number of elements in the array

DEFINE frequency[] // Array to store frequency of each element

DEFINE visited[] // Array to mark elements as visited

DEFINE i, j // Loop variables

DEFINE element // Variable to store current element

// Initialize frequency and visited arrays

DEFINE frequency[size] // To hold frequency counts for each element

DEFINE visited[size] // To mark if an element has been processed

FOR i FROM 0 TO size - 1

frequency[i] = 0

visited[i] = FALSE

END FOR

// Read the number of elements

PRINT "Enter the number of elements:"

READ size

// Read elements into the array

FOR i FROM 0 TO size - 1

PRINT "Enter element ", i + 1, ":"

READ array[i]

END FOR

// Count frequencies of each element

FOR i FROM 0 TO size - 1

IF visited[i] = FALSE THEN

element = array[i]

frequency[i] = 1

// Mark element as visited and count its frequency

FOR j FROM i + 1 TO size - 1

IF array[j] = element THEN

frequency[i] = frequency[i] + 1

visited[j] = TRUE

END IF

END FOR

END IF

END FOR

// Print frequency of each element

PRINT "Element frequencies are:"

FOR i FROM 0 TO size - 1

IF visited[i] = FALSE THEN

PRINT "Element ", array[i], ": Frequency ", frequency[i]

END IF

END FOR

END

12. **Write a Pseudo code to print all unique elements in the array.**

START

// Define the array, its size, and auxiliary structures

DEFINE array[] // The array with elements

DEFINE size // The number of elements in the array

DEFINE unique[] // Array to store unique elements

DEFINE visited[] // Array to mark elements as processed

DEFINE i, j // Loop variables

DEFINE element // Variable to store current element

DEFINE is\_unique // Flag to check if an element is unique

// Initialize the unique and visited arrays

DEFINE unique[size] // To store unique elements

DEFINE visited[size] // To mark elements as processed

FOR i FROM 0 TO size - 1

visited[i] = FALSE

END FOR

// Read the number of elements

PRINT "Enter the number of elements:"

READ size

// Read elements into the array

FOR i FROM 0 TO size - 1

PRINT "Enter element ", i + 1, ":"

READ array[i]

END FOR

// Find unique elements

FOR i FROM 0 TO size - 1

IF visited[i] = FALSE THEN

element = array[i]

is\_unique = TRUE

// Check if element appears more than once

FOR j FROM 0 TO size - 1

IF i != j AND array[j] = element THEN

is\_unique = FALSE

visited[j] = TRUE

END IF

END FOR

// If the element is unique, add it to the unique array

IF is\_unique = TRUE THEN

unique[i] = element

visited[i] = TRUE

END IF

END IF

END FOR

// Print unique elements

PRINT "Unique elements in the array are:"

FOR i FROM 0 TO size - 1

IF unique[i] != NULL THEN

PRINT unique[i]

END IF

END FOR

END

13. **Write a Pseudo code to count total number of duplicate elements in an array.**

START

// Define the array, its size, and auxiliary structures

DEFINE array[] // The array with elements

DEFINE size // The number of elements in the array

DEFINE duplicate\_count // Variable to count duplicate elements

DEFINE visited[] // Array to mark elements as processed

DEFINE i, j // Loop variables

DEFINE element // Variable to store current element

DEFINE is\_duplicate // Flag to check if an element is a duplicate

// Initialize the visited array

DEFINE visited[size] // To mark elements as processed

FOR i FROM 0 TO size - 1

visited[i] = FALSE

END FOR

// Initialize the duplicate count

duplicate\_count = 0

// Read the number of elements

PRINT "Enter the number of elements:"

READ size

// Read elements into the array

FOR i FROM 0 TO size - 1

PRINT "Enter element ", i + 1, ":"

READ array[i]

END FOR

// Count duplicate elements

FOR i FROM 0 TO size - 1

IF visited[i] = FALSE THEN

element = array[i]

is\_duplicate = FALSE

// Check if element appears more than once

FOR j FROM 0 TO size - 1

IF i != j AND array[j] = element THEN

is\_duplicate = TRUE

visited[j] = TRUE // Mark the duplicate occurrences

END IF

END FOR

// If the element is a duplicate, increment the duplicate count

IF is\_duplicate = TRUE THEN

duplicate\_count = duplicate\_count + 1

END IF

// Mark the element itself as visited

visited[i] = TRUE

END IF

END FOR

// Print the count of duplicate elements

PRINT "Total number of duplicate elements:", duplicate\_count

END

14.  **Write a Pseudo code to delete all duplicate elements from an array.**

START

// Define the array, its size, and auxiliary structures

DEFINE array[] // The array with elements

DEFINE size // The number of elements in the array

DEFINE unique\_size // The size of the array after duplicates are removed

DEFINE unique[] // Array to store unique elements

DEFINE visited[] // Array to mark elements as processed

DEFINE i, j // Loop variables

DEFINE element // Variable to store current element

// Initialize the visited array

DEFINE visited[size] // To mark elements as processed

FOR i FROM 0 TO size - 1

visited[i] = FALSE

END FOR

// Initialize the unique size

unique\_size = 0

// Read the number of elements

PRINT "Enter the number of elements:"

READ size

// Read elements into the array

FOR i FROM 0 TO size - 1

PRINT "Enter element ", i + 1, ":"

READ array[i]

END FOR

// Identify unique elements

FOR i FROM 0 TO size - 1

IF visited[i] = FALSE THEN

element = array[i]

// Check if the element appears more than once

IS\_UNIQUE = TRUE

FOR j FROM i + 1 TO size - 1

IF array[j] = element THEN

visited[j] = TRUE // Mark the duplicate occurrence

IS\_UNIQUE = FALSE

END IF

END FOR

// If the element is unique, add it to the unique array

IF IS\_UNIQUE = TRUE THEN

unique[unique\_size] = element

unique\_size = unique\_size + 1

END IF

END IF

END FOR

// Print the updated array with duplicates removed

PRINT "Array after removing duplicates:"

FOR i FROM 0 TO unique\_size - 1

PRINT unique[i]

END FOR

END

15. **Write a Pseudo code to merge two array to third array.**

START

// Define the arrays and their sizes

DEFINE array1[] // First array

DEFINE array2[] // Second array

DEFINE merged[] // Array to store the merged result

DEFINE size1 // Number of elements in the first array

DEFINE size2 // Number of elements in the second array

DEFINE merged\_size // Total number of elements in the merged array

DEFINE i, j // Loop variables

// Read the size and elements of the first array

PRINT "Enter the number of elements in the first array:"

READ size1

DEFINE array1[size1]

PRINT "Enter elements for the first array:"

FOR i FROM 0 TO size1 - 1

PRINT "Enter element ", i + 1, ":"

READ array1[i]

END FOR

// Read the size and elements of the second array

PRINT "Enter the number of elements in the second array:"

READ size2

DEFINE array2[size2]

PRINT "Enter elements for the second array:"

FOR i FROM 0 TO size2 - 1

PRINT "Enter element ", i + 1, ":"

READ array2[i]

END FOR

// Initialize the merged array size

merged\_size = size1 + size2

DEFINE merged[merged\_size]

// Copy elements from the first array to the merged array

FOR i FROM 0 TO size1 - 1

merged[i] = array1[i]

END FOR

// Copy elements from the second array to the merged array

FOR j FROM 0 TO size2 - 1

merged[size1 + j] = array2[j]

END FOR

// Print the merged array

PRINT "Merged array is:"

FOR i FROM 0 TO merged\_size - 1

PRINT merged[i]

END FOR

END

16.  **Write a Pseudo code to find reverse of an array.**

START

// Define the array and its size

DEFINE array[] // The array with elements

DEFINE size // The number of elements in the array

DEFINE start\_index, end\_index // Indices for swapping elements

DEFINE temp // Temporary variable for swapping

DEFINE i // Loop variable

// Read the number of elements

PRINT "Enter the number of elements:"

READ size

// Initialize the array

DEFINE array[size]

PRINT "Enter elements of the array:"

FOR i FROM 0 TO size - 1

PRINT "Enter element ", i + 1, ":"

READ array[i]

END FOR

// Initialize start and end indices

start\_index = 0

end\_index = size - 1

// Reverse the array

WHILE start\_index < end\_index

// Swap elements at start\_index and end\_index

temp = array[start\_index]

array[start\_index] = array[end\_index]

array[end\_index] = temp

// Move indices towards the center

start\_index = start\_index + 1

end\_index = end\_index - 1

END WHILE

// Print the reversed array

PRINT "Reversed array is:"

FOR i FROM 0 TO size - 1

PRINT array[i]

END FOR

END

17.  **Write a Pseudo code to put even and odd elements of array in two separate array.**

START

// Define the arrays and their sizes

DEFINE array[] // The original array with elements

DEFINE even[] // Array to store even elements

DEFINE odd[] // Array to store odd elements

DEFINE size // Number of elements in the original array

DEFINE even\_size // Number of even elements

DEFINE odd\_size // Number of odd elements

DEFINE i // Loop variable

DEFINE element // Variable to store current element

// Initialize sizes for even and odd arrays

even\_size = 0

odd\_size = 0

// Read the number of elements

PRINT "Enter the number of elements:"

READ size

// Initialize the original array

DEFINE array[size]

PRINT "Enter elements of the array:"

FOR i FROM 0 TO size - 1

PRINT "Enter element ", i + 1, ":"

READ array[i]

END FOR

// Initialize the arrays to hold even and odd elements

DEFINE even[SIZE]

DEFINE odd[SIZE]

// Separate even and odd elements

FOR i FROM 0 TO size - 1

element = array[i]

IF element MOD 2 = 0 THEN

// If element is even

even[even\_size] = element

even\_size = even\_size + 1

ELSE

// If element is odd

odd[odd\_size] = element

odd\_size = odd\_size + 1

END IF

END FOR

// Print the even elements

PRINT "Even elements are:"

FOR i FROM 0 TO even\_size - 1

PRINT even[i]

END FOR

// Print the odd elements

PRINT "Odd elements are:"

FOR i FROM 0 TO odd\_size - 1

PRINT odd[i]

END FOR

END

18. **Write a Pseudo code to search an element in an array.**

START

// Define the array, its size, and the target element

DEFINE array[] // The array with elements

DEFINE size // Number of elements in the array

DEFINE target // Element to be searched

DEFINE i // Loop variable

DEFINE found // Boolean flag to indicate if the element is found

// Initialize the found flag

found = FALSE

// Read the number of elements

PRINT "Enter the number of elements:"

READ size

// Initialize the array

DEFINE array[size]

PRINT "Enter elements of the array:"

FOR i FROM 0 TO size - 1

PRINT "Enter element ", i + 1, ":"

READ array[i]

END FOR

// Read the target element to search

PRINT "Enter the element to search for:"

READ target

// Perform linear search

FOR i FROM 0 TO size - 1

IF array[i] = target THEN

found = TRUE

BREAK // Exit the loop if the element is found

END IF

END FOR

// Print the result of the search

IF found = TRUE THEN

PRINT "Element ", target, " found at index ", i

ELSE

PRINT "Element ", target, " not found in the array"

END IF

END

19.  **Write a Pseudo code to sort array elements in ascending or descending order.**

Ascending Order

START

// Define the array and its size

DEFINE array[] // The array with elements

DEFINE size // Number of elements in the array

DEFINE i, j // Loop variables

DEFINE temp // Temporary variable for swapping

// Read the number of elements

PRINT "Enter the number of elements:"

READ size

// Initialize the array

DEFINE array[size]

PRINT "Enter elements of the array:"

FOR i FROM 0 TO size - 1

PRINT "Enter element ", i + 1, ":"

READ array[i]

END FOR

// Bubble Sort in ascending order

FOR i FROM 0 TO size - 2

FOR j FROM 0 TO size - i - 2

IF array[j] > array[j + 1] THEN

// Swap elements

temp = array[j]

array[j] = array[j + 1]

array[j + 1] = temp

END IF

END FOR

END FOR

// Print the sorted array in ascending order

PRINT "Array sorted in ascending order:"

FOR i FROM 0 TO size - 1

PRINT array[i]

END FOR

END

Descending Order :

START

// Define the array and its size

DEFINE array[] // The array with elements

DEFINE size // Number of elements in the array

DEFINE i, j // Loop variables

DEFINE temp // Temporary variable for swapping

// Read the number of elements

PRINT "Enter the number of elements:"

READ size

// Initialize the array

DEFINE array[size]

PRINT "Enter elements of the array:"

FOR i FROM 0 TO size - 1

PRINT "Enter element ", i + 1, ":"

READ array[i]

END FOR

// Bubble Sort in descending order

FOR i FROM 0 TO size - 2

FOR j FROM 0 TO size - i - 2

IF array[j] < array[j + 1] THEN

// Swap elements

temp = array[j]

array[j] = array[j + 1]

array[j + 1] = temp

END IF

END FOR

END FOR

// Print the sorted array in descending order

PRINT "Array sorted in descending order:"

FOR i FROM 0 TO size - 1

PRINT array[i]

END FOR

END

20. **Write a Pseudo code to sort even and odd elements of array separately.**

START

// Define the arrays and their sizes

DEFINE array[] // The original array with elements

DEFINE even[] // Array to store even elements

DEFINE odd[] // Array to store odd elements

DEFINE sorted\_even[] // Array to store sorted even elements

DEFINE sorted\_odd[] // Array to store sorted odd elements

DEFINE size // Number of elements in the original array

DEFINE even\_size // Number of even elements

DEFINE odd\_size // Number of odd elements

DEFINE i, j // Loop variables

DEFINE temp // Temporary variable for swapping

// Initialize sizes for even and odd arrays

even\_size = 0

odd\_size = 0

// Read the number of elements

PRINT "Enter the number of elements:"

READ size

// Initialize the original array

DEFINE array[size]

PRINT "Enter elements of the array:"

FOR i FROM 0 TO size - 1

PRINT "Enter element ", i + 1, ":"

READ array[i]

END FOR

// Separate even and odd elements

FOR i FROM 0 TO size - 1

IF array[i] MOD 2 = 0 THEN

// Element is even

even[even\_size] = array[i]

even\_size = even\_size + 1

ELSE

// Element is odd

odd[odd\_size] = array[i]

odd\_size = odd\_size + 1

END IF

END FOR

// Define arrays for sorted even and odd elements

DEFINE sorted\_even[even\_size]

DEFINE sorted\_odd[odd\_size]

// Copy elements to sorted arrays

FOR i FROM 0 TO even\_size - 1

sorted\_even[i] = even[i]

END FOR

FOR i FROM 0 TO odd\_size - 1

sorted\_odd[i] = odd[i]

END FOR

// Sort even elements in ascending order

FOR i FROM 0 TO even\_size - 2

FOR j FROM 0 TO even\_size - i - 2

IF sorted\_even[j] > sorted\_even[j + 1] THEN

// Swap elements

temp = sorted\_even[j]

sorted\_even[j] = sorted\_even[j + 1]

sorted\_even[j + 1] = temp

END IF

END FOR

END FOR

// Sort odd elements in ascending order

FOR i FROM 0 TO odd\_size - 2

FOR j FROM 0 TO odd\_size - i - 2

IF sorted\_odd[j] > sorted\_odd[j + 1] THEN

// Swap elements

temp = sorted\_odd[j]

sorted\_odd[j] = sorted\_odd[j + 1]

sorted\_odd[j + 1] = temp

END IF

END FOR

END FOR

// Merge sorted even and odd arrays back into the original array

DEFINE merged\_size = even\_size + odd\_size

DEFINE merged[merged\_size]

FOR i FROM 0 TO even\_size - 1

merged[i] = sorted\_even[i]

END FOR

FOR j FROM 0 TO odd\_size - 1

merged[even\_size + j] = sorted\_odd[j]

END FOR

// Print the sorted even elements

PRINT "Sorted even elements are:"

FOR i FROM 0 TO even\_size - 1

PRINT sorted\_even[i]

END FOR

// Print the sorted odd elements

PRINT "Sorted odd elements are:"

FOR i FROM 0 TO odd\_size - 1

PRINT sorted\_odd[i]

END FOR

// Print the merged array

PRINT "Merged array with sorted even and odd elements is:"

FOR i FROM 0 TO merged\_size - 1

PRINT merged[i]

END FOR

END