1. #include <stdio.h>

void swap(int x, int y) {

int temp = x;

x = y;

y = temp;

printf("After swapping in function (call by value): x = %d, y = %d\n", x, y);

}

int main() {

int a = 10, b = 20;

printf("Before swapping in main: a = %d, b = %d\n", a, b);

swap(a, b); // Values of a and b are copied to x and y

printf("After swapping in main (call by value): a = %d, b = %d\n", a, b);

return 0;

}

2.#include <stdio.h>

int \*findDuplicates(int arr[], int n) {

int \*duplicates = (int \*)malloc(n \* sizeof(int)); // Allocate memory for potential duplicates

int duplicatesCount = 0;

for (int i = 0; i < n; i++) {

int index = arr[i] % n; // Map elements to indices within array bounds

arr[index] += n; // Mark occurrences by incrementing values at corresponding indices

}

for (int i = 0; i < n; i++) {

if (arr[i] / n > 1) { // Check for elements that have been marked more than once

duplicates[duplicatesCount++] = i; // Store the original value (index)

}

}

if (duplicatesCount == 0) {

duplicates[0] = -1; // No duplicates found, return [-1]

3.#include <stdio.h>

void printUnion(int arr1[], int arr2[], int n, int m) {

int i = 0, j = 0;

while (i < n && j < m) {

if (arr1[i] < arr2[j]) {

printf("%d ", arr1[i++]);

} else if (arr2[j] < arr1[i]) {

printf("%d ", arr2[j++]);

} else { // Elements are equal

printf("%d ", arr1[i]); // Print only once

i++;

j++;

}

}

// Print remaining elements from either array

while (i < n) {

printf("%d ", arr1[i++]);

}

while (j < m) {

printf("%d ", arr2[j++]);

}

}

int main() {

int n = 5, arr1[] = {2, 2, 3, 4, 5};

int m = 5, arr2[] = {1, 1, 2, 3, 4};

printf("Union of two arrays is:\n");

printUnion(arr1, arr2, n, m);

return 0;

}