**1.Swapping of two Numbers** by   
a)Call By Value  
b)Call By Reference

**SWAP BY VALUE :**

**CODE :**

#include <stdio.h>

void swapByValue(int x, int y) {

int temp;

temp = x;

x = y;

y = temp;

}

int main() {

int a = 10, b = 20;

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

swapByValue(a, b);

printf("After swapping using Call By Value: a = %d, b = %d\n", a, b);

return 0;

}

**SWAP BY REFERENCE :**

**CODE :**

#include <stdio.h>

void swapByReference(int \*x, int \*y) {

int temp;

temp = \*x;

\*x = \*y;

\*y = temp;

}

int main() {

int a = 10, b = 20;

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

swapByReference(&a, &b);

printf("After swapping using Call By Reference: a = %d, b = %d\n", a, b);

return 0;

}

**2.Find duplicates in an array**

**Given an array a of size N which contains elements from 0 to N-1, you need to find all the elements occurring more than once in the given array. Return the answer in ascending order. If no such element is found, return list containing [-1].**

**CODE :**

#include <stdio.h>

int main() {

int N = 5;

int a[] = {2, 3, 1, 2, 3};

int returnSize = 0;

int counts[N];

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

counts[i] = 0;

}

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

counts[a[i]]++;

}

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

if (counts[i] > 1) {

returnSize++;

}

}

if (returnSize == 0) {

printf("-1\n");

} else {

printf("Duplicates: ");

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

if (counts[a[i]] > 1) {

printf("%d ", a[i]);

counts[a[i]] = 0;

}

}

printf("\n");

}

return 0;

}

**3.Union of Two Sorted Arrays**

**Union of two arrays can be defined as the common and distinct elements in the two arrays. Given two sorted arrays of size n and m respectively, find their union.**

**CODE :**

#include <stdio.h>

int main() {

int n = 5;

int arr1[] = {1, 2, 3, 4, 5};

int m = 3;

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

int i = 0, j = 0;

printf("Union of arrays: ");

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 {

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

j++;

}

}

while (i < n) {

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

}

while (j < m) {

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

}

printf("\n");

return 0;

}