

#### 4. Implement a C Program for Given an array finding duplication values

Code

```
#include <stdio.h>

int main() {
    int a[] = {1, 2, 3};
    int b[] = {2, 3, 4};
    int merged[10];
    int n1 = 3, n2 = 3;
    int i, j, k = 0;

    // Merge arrays
    for (i = 0; i < n1; i++)
        merged[k++] = a[i];
    for (j = 0; j < n2; j++)
        merged[k++] = b[j];

    // Find duplicates
    printf("Duplicate elements are: ");
    int found = 0;
    for (i = 0; i < n1 + n2; i++) {
        for (j = i + 1; j < n1 + n2; j++) {
            if (merged[i] == merged[j]) {
                printf("%d ", merged[i]);
                found = 1;
                break;
            }
        }
    }
}
```

```
}
```

```
if (!found)
```

```
    printf("None");
```

```
printf("\n");
```

```
return 0;
```

```
}
```

## Output

```
main.c
1 #include <stdio.h>
2
3 int main() {
4     int a[] = {1, 2, 3};
5     int b[] = {2, 3, 4};
6     int merged[10];
7     int n1 = 3, n2 = 3;
8     int i, j, k = 0;
9
10    // Merge arrays
11    for (i = 0; i < n1; i++)
12        merged[k++] = a[i];
13    for (j = 0; j < n2; j++)
14        merged[k++] = b[j];
15
16    // Find duplicates
17    printf("Duplicate elements are: ");
18    int found = 0;
19    for (i = 0; i < n1 + n2; i++) {
20        for (j = i + 1; j < n1 + n2; j++) {
21            if (merged[i] == merged[j]) {
22                printf("%d ", merged[i]);
23                found = 1;
24                break;
25            }
26        }
27    }
28
29    if (!found)
30        printf("None");
31
32    printf("\n");
33    return 0;
34 }
```

Output

Duplicate elements are: 2 3

=== Code Execution Successful ===