

Practice Problem:

Array

Problem 1: Find the minimum and maximum element in an array.

Solve:

```
#include <stdio.h>
```

```
void findMinMax(int arr[], int n)
```

```
{
```

```
    int min = arr[0];
```

```
    int max = arr[0];
```

```
    for (int i = 1; i < n; i++)
```

```
    {
```

```
        if (arr[i] < min)
```

```
            min = arr[i];
```

```
        else if (arr[i] > max)
```

```
            max = arr[i];
```

```
    }
```

```
    printf("min: %d max: %d\n", min, max);
```

```
}
```

```
int main()
```

```
{
```

```
    int arr[] = {5, 2, 7, 10, 1, 9, 25, 20};
```

```
    int n = sizeof(arr) / sizeof(arr[0]);
```

```
    findMinMax(arr, n);
```

```
    return 0;
```

```
}
```

Problem 2: Write a program to reverse the array.

Solve:

```
#include <stdio.h>
```

```
void reverse(int array[], int size)
```

```
{
```

```
    int r[size];
```

```
    for (int i = 0; i < size; i++)
```

```
    {
```

```
        r[i] = array[size - i - 1];
```

```
    }
```

```
    printf("Reversed Array: ");
```

```
    for (int i = 0; i < size; i++)
```

```
    {
```

```
        printf("%d ", r[i]);
```

```
    }
```

```
}
```

```
int main()
```

```
{
```

```
    int array[] = {19, 23, 13, 23, 14};
```

```
    int size = sizeof(array) / sizeof(array[0]);
```

```
    reverse(array, size);
```

```
    return 0;
```

```
}
```

Problem 3: Write a Program to cyclically rotate an array by one.

Solve:

```
#include <stdio.h>
```

```
void r(int arr[], int n)
```

```
{
```

```
    int last = arr[n - 1];
```

```
    for (int i = n - 1; i > 0; i--)
```

```
    {
```

```
        arr[i] = arr[i - 1];
```

```
    }
```

```
    arr[0] = last;
```

```
}
```

```
void printArray(int arr[], int n)
```

```
{
```

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

```
    {
```

```
        printf("%d ", arr[i]);
```

```
    }
```

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

```
}
```

```
int main()
{
    int arr[] = {1, 2, 3, 4, 5};
    int n = sizeof(arr) / sizeof(arr[0]);

    printf("Original array: \n");
    printArray(arr, n);

    r(arr, n);

    printf("Array after rotation: \n");
    printArray(arr, n);

    return 0;
}
```

Problem 4: Write a program to sort the given array.

Solve:

```
#include <stdio.h>
```

```
void Sort(int array[], int n)
```

```
{
```

```
    for (int i = 0; i < n - 1; i++)
```

```
    {
```

```
        for (int j = 0; j < n - i - 1; j++)
```

```
        {
```

```
            if (array[j] > array[j + 1])
```

```
            {
```

```
                int t = array[j];
```

```
                array[j] = array[j + 1];
```

```
                array[j + 1] = t;
```

```
            }
```

```
        }
```

```
    }
```

```
}
```

```
int main()
```

```
{
```

```
    int array[] = {30, 50, 20, 19, 7, 22};
```

```
    int n = sizeof(array) / sizeof(array[0]);
```

```
    Sort(array, n);
```

```
for (int i = 0; i < n; i++)  
{  
    printf("%d ", array[i]);  
}  
return 0;  
}
```

Problem 5: Find duplicate elements in an array.

Solve:

```
#include <stdio.h>
```

```
void find(int arr[], int n)
```

```
{
```

```
    int found = 0;
```

```
    printf("Duplicate elements in the array are: \n");
```

```
    for (int i = 0; i < n - 1; i++)
```

```
    {
```

```
        for (int j = i + 1; j < n; j++)
```

```
        {
```

```
            if (arr[i] == arr[j])
```

```
            {
```

```
                printf("%d ", arr[i]);
```

```
                found = 1;
```

```
                break;
```

```
            }
```

```
        }
```

```
    }
```

```
    if (!found)
```

```
    {
```

```
        printf("No duplicates found");
```

```
    }
```



```
}
```

```
int main()
```

```
{
```

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

```
    int n = sizeof(arr) / sizeof(arr[0]);
```

```
    find(arr, n);
```

```
    return 0;
```

```
}
```

Problem 6: Find the occurrence of an integer in the array.

Solve:

```
#include <stdio.h>
```

```
int main()
```

```
{
```

```
    int n, t, count = 0;
```

```
    printf("Enter the number of elements in the array: ");
```

```
    scanf("%d", &n);
```

```
    int arr[n];
```

```
    printf("Enter the elements of the array: \n");
```

```
for (int i = 0; i < n; i++)
{
    scanf("%d", &arr[i]);
}

printf("Enter the number to find its occurrence: ");
scanf("%d", &t);

for (int i = 0; i < n; i++)
{
    if (arr[i] == t)
    {
        count++;
    }
}

if (count > 0)
{
    printf("The number %d occurs %d time in the array.\n", t, count);
}
else
{
    printf("The number %d does not occur in the array.\n", t);
}

return 0;
}
```

Problem 7: Sort the array of 0s, 1s, and 2s.

Solve:

```
#include <stdio.h>
```

```
void sort012(int arr[], int n) {
```

```
    int low = 0, mid = 0, high = n - 1;
```

```
    while (mid <= high) {
```

```
        switch (arr[mid]) {
```

```
            case 0:
```

```
            {
```

```
                int temp = arr[low];
```

```
                arr[low] = arr[mid];
```

```
                arr[mid] = temp;
```

```
            }
```

```
            low++;
```

```
            mid++;
```

```
            break;
```

```
            case 1:
```

```
                mid++;
```

```
                break;
```

```
            case 2:
```

```
                // Swap arr[mid] and arr[high]
```

```
                {
```

```
                    int temp = arr[mid];
```

```
                    arr[mid] = arr[high];
```

```
                    arr[high] = temp;
```

```
    }  
    high--;  
    break;  
}  
}  
}
```

```
void printArray(int arr[], int n) {  
    for (int i = 0; i < n; i++) {  
        printf("%d ", arr[i]);  
    }  
    printf("\n");  
}
```

```
int main() {  
    int arr[] = {0, 1, 2, 0, 1, 2, 1, 0};  
    int n = sizeof(arr) / sizeof(arr[0]);  
  
    sort012(arr, n);  
    printArray(arr, n);  
  
    return 0;  
}
```

Problem 8: Move all the negative element to one side of the array.

Solve:

```
#include <stdio.h>
```

```
void move(int arr[], int n)
```

```
{
```

```
    int j = 0;
```

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

```
    {
```

```
        if (arr[i] < 0)
```

```
        {
```

```
            if (i != j)
```

```
            {
```

```
                int temp = arr[i];
```

```
                arr[i] = arr[j];
```

```
                arr[j] = temp;
```

```
            }
```

```
            j++;
```

```
        }
```

```
    }
```

```
}
```

```
void printArray(int arr[], int n)
```

```
{
```

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

```
    {  
        printf("%d ", arr[i]);  
    }  
    printf("\n");  
}
```

```
int main()  
{  
    int arr[] = {1, -1, 3, 2, -7, -5, 11, 6};  
    int n = sizeof(arr) / sizeof(arr[0]);  
  
    move(arr, n);  
    printArray(arr, n);  
  
    return 0;  
}
```

Problem 9: Find the row with maximum number of 1s.

Solve:

```
#include <stdio.h>
```

```
int main()
```

```
{
```

```
    int mat[4][4] = {{0, 1, 1, 1},  
                      {0, 0, 1, 1},  
                      {1, 1, 1, 1},  
                      {0, 0, 0, 0}};
```

```
    int max_count = 0, index = -1;
```

```
    for (int i = 0; i < 4; i++)
```

```
    {
```

```
        int count = 0;
```

```
        for (int j = 0; j < 4; j++)
```

```
        {
```

```
            if (mat[i][j] == 1)
```

```
                count++;
```

```
        }
```

```
        if (count > max_count)
```

```
        {
```

```
            max_count = count;
```

```
            index = i;
```

```
        }
```

```
}

printf("Index of row with maximum 1s is %d", index);
}
```

Problem 10: Find the majority element of an array.

Solve:

```
#include <stdio.h>
```

```
int major(int arr[], int n)
{

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

        for (int j = 0; j < n; j++)
        {
            if (arr[i] == arr[j])
            {
                count++;
            }
        }

        if (count > n / 2)
        {
            return arr[i];
        }
    }
}
```



```

    }
}

return -1;
}

int main()
{
    int arr[] = {5, 3, 5, 4, 5, 5};
    int n = sizeof(arr) / sizeof(arr[0]);

    printf("%d\n", major(arr, n));

    return 0;
}

```

Problem 11: Sort an array in wave form.

Solve:

```

#include <stdio.h>

int main()
{
    int array[] = {10, 49, 2, 1, 5, 23};

    int n = sizeof(array) / sizeof(array[0]);
    int temp;

    // step 1
    for (int i = 0; i < n - 1; i++)

```

```

{
    for (int j = i + 1; j < n; j++)
    {
        if (array[i] > array[j])
        {
            temp = array[i];
            array[i] = array[j];
            array[j] = temp;
        }
    }
}

// step 2
for (int i = 0; i < n; i = i + 2)
{
    temp = array[i];
    array[i] = array[i + 1];
    array[i + 1] = temp;
}

for (int i = 0; i < n; i++)
{
    printf("%d ", array[i]);
}

return 0;
}

```

Problem 12: Find a peak element which is not smaller than its neighbors.

Solve:

```
#include <stdio.h>

int findPeak(int arr[], int n)
{
    int low = 0, high = n - 1;

    while (low <= high)
    {
        int mid = low + (high - low) / 2;

        if ((mid == 0 || arr[mid - 1] <= arr[mid]) &&
            (mid == n - 1 || arr[mid + 1] <= arr[mid]))
        {
            return mid;
        }

        else if (mid > 0 && arr[mid - 1] > arr[mid])
        {
            high = mid - 1;
        }

        else
        {
            low = mid + 1;
        }
    }
}
```

```
}

return -1;
}

int main()
{
    int arr[] = {5, 10, 20, 15};
    int n = sizeof(arr) / sizeof(arr[0]);

    int peakIndex = findPeak(arr, n);

    if (peakIndex != -1)
    {
        printf("A peak element is found at index %d, with value %d\n", peakIndex, arr[peakIndex]);
    }
    else
    {
        printf("No peak element found.\n");
    }

    return 0;
}
```

Problem 13: Find the Kth largest and Kth smallest number in an array.

Solve:

```
#include <stdio.h>

void sort(int arr[], int n)
{
    int temp;
    for (int i = 0; i < n - 1; i++)
    {
        for (int j = i + 1; j < n; j++)
        {
            if (arr[i] > arr[j])
            {
                temp = arr[i];
                arr[i] = arr[j];
                arr[j] = temp;
            }
        }
    }
}

int main()
{
    int n, k;

    printf("Enter the number of elements in the array: ");
    scanf("%d", &n);
```

```
int arr[n];
```

```
printf("Enter the elements of the array: \n");
```

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

```
{
```

```
    scanf("%d", &arr[i]);
```

```
}
```

```
printf("Enter the value of K: ");
```

```
scanf("%d", &k);
```

```
sort(arr, n);
```

```
printf("The %dth smallest element is: %d\n", k, arr[k - 1]);
```

```
printf("The %dth largest element is: %d\n", k, arr[n - k]);
```

```
return 0;
```

```
}
```