ASSIGNMENT-8

1. Find minimum and maximum number in array.

```
#include <stdio.h>
void findMinAndMax(int array[], int size, int* min, int* max) {
 *min = array[0];
 *max = array[0];
 for (int i = 1; i < size; i++) {
 if (array[i] < *min) {
 *min = array[i];
 } else if (array[i] > *max) {
 *max = array[i];
 }
 }
}
int main() {
 int array[] = {12, 45, 7, 23, 56, 89, 34};
 int size = sizeof(array) / sizeof(array[0]);
 int min, max;
 findMinAndMax(array, size, &min, &max);
 printf("Min: %d, Max: %d\n", min, max);
 return 0;
}
```

2. Search the given number in array.

```
#include <stdio.h>
int searchNumber(int array[], int size, int target) {
 for (int i = 0; i < size; i++) {
 if (array[i] == target) {
 return i; // Return the index of the target number
 }
 }
 return -1; // Return -1 if the target number is not found
}
int main() {
 int array[] = {12, 45, 7, 23, 56, 89, 34};
 int size = sizeof(array) / sizeof(array[0]);
 int target = 23;
 int result = searchNumber(array, size, target);
 if (result != -1) {
 printf("Target number %d found at index %d\n", target, result);
 } else {
 printf("Target number %d not found in the array\n", target)
}
 return 0;
}
```

3. Find sum of all numbers.

```
#include <stdio.h>
int sumOfArray(int array[], int size) {
 int sum = 0;
 for (int i = 0; i < size; i++) {
 sum += array[i];
 }
 return sum;
}
int main() {
 int array[] = {12, 45, 7, 23, 56, 89, 34};
 int size = sizeof(array) / sizeof(array[0]);
 int sum = sumOfArray(array, size);
 printf("Sum of all numbers in the array: %d\n", sum);
 return 0;
}
```

4. Find odd and even among the numbers.

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

```
void findOddAndEven(int array[], int size) {
 printf("Even numbers: ");
 for (int i = 0; i < size; i++) {
 if (array[i] % 2 == 0) {
 printf("%d ", array[i]);
 }
 }
 printf("\nOdd numbers: ");
 for (int i = 0; i < size; i++) {
 if (array[i] % 2 != 0) {
 printf("%d ", array[i]);
 }
 }
}
int main() {
 int array[] = {12, 45, 7, 23, 56, 89, 34};
 int size = sizeof(array) / sizeof(array[0]);
 findOddAndEven(array, size);
 return 0;
```

5. Print alternate elements in array.

```
#include <stdio.h>
void printAlternateElements(int array[], int size) {
 printf("Alternate elements: ");
 for (int i = 0; i < size; i += 2) {
 printf("%d ", array[i]);
 }
}
int main() {
 int array[] = {12, 45, 7, 23, 56, 89, 34};
 int size = sizeof(array) / sizeof(array[0]);
 printAlternateElements(array, size);
 return 0;
}
```

6. Accept array and print only prime numbers of array.

```
#include <stdio.h>
// Function to check if a number is prime
int isPrime(int num) {
 if (num <= 1) return 0;
 for (int i = 2; i * i <= num; i++) {
 if (num % i == 0) return 0;
 }
 return 1;
}
// Function to print prime numbers in the array
void printPrimeNumbers(int array[], int size) {
 printf("Prime numbers: ");
 for (int i = 0; i < size; i++) {
 if (isPrime(array[i])) {
 printf("%d ", array[i]);
 }}
}
int main() {
 int array[] = {12, 45, 7, 23, 56, 89, 34};
 int size = sizeof(array) / sizeof(array[0]);
 printPrimeNumbers(array, size);
 return 0;
}
```

7. Take two array and add sum in third array

```
#include <stdio.h>
// Function to add two arrays and store the sum in a third array
void addArrays(int arr1[], int arr2[], int result[], int size) {
 for (int i = 0; i < size; i++) {
```

```
 result[i] = arr1[i] + arr2[i];
 }
}
int main() {
 int arr1[5] = {1, 2, 3, 4, 5};
 int arr2[5] = {10, 20, 30, 40, 50};
 int result[5];
 addArrays(arr1, arr2, result, 5);
 printf("Resultant array: ");
 for (int i = 0; i < 5; i++) {
 printf("%d ", result[i]);
 }
 return 0;
}
```

8. Merge two arrays

```
#include <stdio.h>
// Function to merge two arrays
void mergeArrays(int arr1[], int arr2[], int result[], int size1, int size2) {
 int i = 0, j = 0, k = 0;
 // Merge smaller elements first
 while (i < size1 && j < size2) {
 if (arr1[i] < arr2[j]) {
 result[k++] = arr1[i++];
 } else {
 result[k++] = arr2[j++];
 }
 }
 // Copy remaining elements of arr1, if any
 while (i < size1) {
 result[k++] = arr1[i++];
 }
 // Copy remaining elements of arr2, if any
 while (j < size2) {
 result[k++] = arr2[j++];
 }
}
int main() {
 int arr1[5] = {1, 3, 5, 7, 9};
 int arr2[5] = {2, 4, 6, 8, 10};
 int result[10];
 mergeArrays(arr1, arr2, result, 5, 5);
 printf("Merged array: ");
 for (int i = 0; i < 10; i++) {
 printf("%d ", result[i]);
 }
 return 0;
}
```

9. Reverse the given array.

```
#include <stdio.h>
// Function to reverse an array
void reverseArray(int arr[], int size) {
 int temp, start = 0;
 int end = size - 1;
 while (start < end) {
 temp = arr[start];
 arr[start] = arr[end];
 arr[end] = temp;
 start++;
 end--;
 }
}
int main() {
 int arr[5] = {1, 2, 3, 4, 5};
 printf("Original array: ");
 for (int i = 0; i < 5; i++) {
 printf("%d ", arr[i]);
 }
 reverseArray(arr, 5);
 printf("\nReversed array: ");
 for (int i = 0; i < 5; i++) {
 printf("%d ", arr[i]);
 }
 return 0;
}
```

10. Sort the array.

```
#include <stdio.h>
// Function to swap two elements
void swap(int* a, int* b) {
 int temp = *a;
 *a = *b;
 *b = temp;
}
// Function to sort an array using Bubble Sort
void sortArray(int arr[], int size) {
 for (int i = 0; i < size - 1; i++) {
 for (int j = 0; j < size - i - 1; j++) {
 if (arr[j] > arr[j + 1]) {
 swap(&arr[j], &arr[j + 1]);
 }
 }
 }
}
int main() {
 int arr[5] = {5, 2, 8, 1, 9};
 printf("Original array: ");
 for (int i = 0; i < 5; i++) {
 printf("%d ", arr[i]);
 }
 sortArray(arr, 5);
 printf("\nSorted array: ");
 for (int i = 0; i < 5; i++) {
 printf("%d ", arr[i]);
 }
 return 0;
}
```