

## 1. Matrix Addition

```
#include <stdio.h>
#include<conio.h>
void main() {
    int rows, cols;
    printf("Enter the number of rows and columns: ");
    scanf("%d %d", &rows, &cols);
    int matrix1[rows][cols], matrix2[rows][cols],    result[rows][cols];
    printf("Enter elements of first matrix:\n");
    for (int i = 0; i < rows; i++)
        for (int j = 0; j < cols; j++)
            scanf("%d", &matrix1[i][j]);
    printf("Enter elements of second matrix:\n");
    for (int i = 0; i < rows; i++)
        for (int j = 0; j < cols; j++)
            scanf("%d", &matrix2[i][j]);
    // Matrix Addition
    for (int i = 0; i < rows; i++)
        for (int j = 0; j < cols; j++)
            result[i][j] = matrix1[i][j] + matrix2[i][j];

    printf("Resultant Matrix after Addition:\n");
    for (int i = 0; i < rows; i++) {
        for (int j = 0; j < cols; j++)
            printf("%d ", result[i][j]);
        printf("\n");
    }
```

```
}  
}
```

## 2. Matrix Subtraction

```
#include <stdio.h>  
  
void main() {  
    int rows, cols;  
    printf("Enter the number of rows and columns: ");  
    scanf("%d %d", &rows, &cols);  
    int matrix1[rows][cols], matrix2[rows][cols],    result[rows][cols];  
    printf("Enter elements of first matrix:\n");  
    for (int i = 0; i < rows; i++)  
        for (int j = 0; j < cols; j++)  
            scanf("%d", &matrix1[i][j]);  
    printf("Enter elements of second matrix:\n");  
    for (int i = 0; i < rows; i++)  
        for (int j = 0; j < cols; j++)  
            scanf("%d", &matrix2[i][j]);  
    // Matrix Subtraction  
    for (int i = 0; i < rows; i++)  
        for (int j = 0; j < cols; j++)  
            result[i][j] = matrix1[i][j] - matrix2[i][j];  
    printf("Resultant Matrix after Subtraction:\n");  
    for (int i = 0; i < rows; i++) {  
        for (int j = 0; j < cols; j++)  
            printf("%d ", result[i][j]);  
        printf("\n");  
    }  
}
```

```
}
```

### 3. Matrix Multiplication

```
#include <stdio.h>

void main() {
    int rows1, cols1, rows2, cols2;
    printf("Enter rows and columns for the first matrix:");
    scanf("%d %d", &rows1, &cols1);
    printf("Enter rows and columns for the second matrix: ");
    scanf("%d %d", &rows2, &cols2);
    if (cols1 != rows2) {
        printf("Matrix multiplication is not possible.\n");
        return;
    }
    int matrix1[rows1][cols1], matrix2[rows2][cols2], result[rows1][cols2];
    printf("Enter elements of first matrix:\n");

    for (int i = 0; i < rows1; i++)
        for (int j = 0; j < cols1; j++)
            scanf("%d", &matrix1[i][j]);
    printf("Enter elements of second matrix:\n");
```

```

for (int i = 0; i < rows2; i++)
    for (int j = 0; j < cols2; j++)
        scanf("%d", &matrix2[i][j]);
// Initialize result matrix to 0
for (int i = 0; i < rows1; i++)
    for (int j = 0; j < cols2; j++)
        result[i][j] = 0;
// Matrix Multiplication
for (int i = 0; i < rows1; i++)
    for (int j = 0; j < cols2; j++)
        for (int k = 0; k < cols1; k++)
            result[i][j] += matrix1[i][k] * matrix2[k][j];
printf("Resultant Matrix after Multiplication:\n");
for (int i = 0; i < rows1; i++) {
    for (int j = 0; j < cols2; j++)
        printf("%d ", result[i][j]);
    printf("\n");
}
}

```

#### 4. Removing duplicate elements in an array

```

#include <stdio.h>

int main() {
    int n, i, j, k;

    // Input: Array size

```

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

int array[n]; // Declare the array

// Input: Array elements
printf("Enter %d elements:\n", n);
for (i = 0; i < n; i++) {
    scanf("%d", &array[i]);
}

// Processing: Remove duplicates
for (i = 0; i < n; i++) {
    for (j = i + 1; j < n; j++) {
        if (array[i] == array[j]) {
            // Shift elements to the left to overwrite duplicate
            for (k = j; k < n - 1; k++) {
                array[k] = array[k + 1];
            }
            n--; // Reduce array size
            j--; // Adjust the index to check the new element at position j
        }
    }
}

// Output: Array after removing duplicates
printf("Array after removing duplicates:\n");
```

```

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

return 0;
}

```

## 5. CHECK IF AN ARRAY IS PALINDROMIC

```

#include <stdio.h>

int main() {
    int n, isPalindrome = 1;
    // Input: Size of the array
    printf("Enter the number of elements: ");
    scanf("%d", &n);
    int array[n]; // Declare a one-dimensional array
    // Input: Array elements
    printf("Enter %d elements:\n", n);
    for (int i = 0; i < n; i++) {
        scanf("%d", &array[i]);
    }
    // Check if the array is palindromic
    for (int i = 0; i < n / 2; i++) {
        if (array[i] != array[n - i - 1]) {
            isPalindrome = 0; // Array is not palindromic
            break;
        }
    }
}

```

```
    }  
}  
  
// Output the result  
if (isPalindrome) {  
    printf("The array is palindromic.\n");  
} else {  
    printf("The array is not palindromic.\n");  
}  
  
return 0;  
}
```