Deletion of element at a specific position in a array:

```
 #include <stdio.h>

#define MAX_SIZE 100

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

void deleteElement(int arr[], int *size, int position) {
    if (position >= *size || position < 0) {
        printf("Invalid position!\n");
        return;
    }

    int i;
    for (i = position; i < *size - 1; i++) {
        arr[i] = arr[i + 1];
    }
    (*size)--;
}

int main() {
    int arr[MAX_SIZE];
    int size, i, position;

    printf("Enter the size of the array (up to %d): ", MAX_SIZE);
    scanf("%d", &size);

    if (size <= 0 || size > MAX_SIZE) {
        printf("Invalid size!\n");
        return 0;
    }

    printf("Enter the elements of the array:\n");
    for (i = 0; i < size; i++) {
        scanf("%d", &arr[i]);
    }

    printf("Enter the position of the element to delete (0-%d): ", size - 1);
    scanf("%d", &position);

    deleteElement(arr, &size, position);
    printf("Array after deletion: ");
    displayArray(arr, size);

    return 0;
}
```

Smallest element in an array

```
#include <stdio.h>

#define MAX_SIZE 100

int findSmallestElement(int arr[], int size) {
    int smallest = arr[0];
    int i;

    for (i = 1; i < size; i++) {
        if (arr[i] < smallest) {
            smallest = arr[i];
        }
    }

    return smallest;
}

int main() {
    int arr[MAX_SIZE];
    int size, i;

    printf("Enter the size of the array (up to %d): ", MAX_SIZE);
    scanf("%d", &size);

    if (size <= 0 || size > MAX_SIZE) {
        printf("Invalid size!\n");
        return 0;
    }

    printf("Enter the elements of the array:\n");
    for (i = 0; i < size; i++) {
        scanf("%d", &arr[i]);
    }

    int smallest = findSmallestElement(arr, size);
    printf("The smallest element in the array is: %d\n", smallest);

    return 0;
}
```

Sum of elements of an array

```
#include <stdio.h>

#define MAX_SIZE 100

int calculateSum(int arr[], int size) {
    int sum = 0;
    int i = 0;

    while (i < size) {
        sum += arr[i];
        i++;
    }

    return sum;
}

int main() {
    int arr[MAX_SIZE];
    int size, i;

    printf("Enter the size of the array (up to %d): ", MAX_SIZE);
    scanf("%d", &size);

    if (size <= 0 || size > MAX_SIZE) {
        printf("Invalid size!\n");
        return 0;
    }

    printf("Enter the elements of the array:\n");
    for (i = 0; i < size; i++) {
        scanf("%d", &arr[i]);
    }

    int sum = calculateSum(arr, size);
    printf("The sum of elements in the array is: %d\n", sum);

    return 0;
}
```

Matrix addition

```
#include <stdio.h>

#define MAX_SIZE 10
```

```
void matrixAddition(int mat1[][MAX_SIZE], int mat2[][MAX_SIZE], int result[][MAX_SIZE], int
rows, int columns) {
    int i, j;

    for (i = 0; i < rows; i++) {
        for (j = 0; j < columns; j++) {
            result[i][j] = mat1[i][j] + mat2[i][j];
        }
    }
}

void displayMatrix(int mat[][MAX_SIZE], int rows, int columns) {
    int i, j;

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

int main() {
    int mat1[MAX_SIZE][MAX_SIZE];
    int mat2[MAX_SIZE][MAX_SIZE];
    int result[MAX_SIZE][MAX_SIZE];
    int rows, columns, i, j;

    printf("Enter the number of rows (up to %d): ", MAX_SIZE);
    scanf("%d", &rows);

    printf("Enter the number of columns (up to %d): ", MAX_SIZE);
    scanf("%d", &columns);

    printf("Enter the elements of the first matrix:\n");
    for (i = 0; i < rows; i++) {
        for (j = 0; j < columns; j++) {
            scanf("%d", &mat1[i][j]);
        }
    }

    printf("Enter the elements of the second matrix:\n");
    for (i = 0; i < rows; i++) {
        for (j = 0; j < columns; j++) {
            scanf("%d", &mat2[i][j]);
        }
    }
    matrixAddition(mat1, mat2, result, rows, columns);

    printf("Resultant matrix after addition:\n");
    displayMatrix(result, rows, columns);

    return 0;
}
```

Matrix multiplication

```
#include <stdio.h>

#define MAX_SIZE 10

void matrixMultiplication(int mat1[][MAX_SIZE], int mat2[][MAX_SIZE], int
result[][MAX_SIZE], int rows1, int columns1, int columns2) {
    int i, j, k;

    for (i = 0; i < rows1; i++) {
        for (j = 0; j < columns2; j++) {
            result[i][j] = 0;
            for (k = 0; k < columns1; k++) {
                result[i][j] += mat1[i][k] * mat2[k][j];
            }
        }
    }
}

void displayMatrix(int mat[][MAX_SIZE], int rows, int columns) {
    int i, j;

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

int main() {
    int mat1[MAX_SIZE][MAX_SIZE];
    int mat2[MAX_SIZE][MAX_SIZE];
    int result[MAX_SIZE][MAX_SIZE];
    int rows1, columns1, rows2, columns2, i, j;

    printf("Enter the number of rows for the first matrix (up to %d): ", MAX_SIZE);
    scanf("%d", &rows1);

    printf("Enter the number of columns for the first matrix (up to %d): ", MAX_SIZE);
    scanf("%d", &columns1);

    printf("Enter the number of rows for the second matrix (up to %d): ", MAX_SIZE);
    scanf("%d", &rows2);

    printf("Enter the number of columns for the second matrix (up to %d): ", MAX_SIZE);
    scanf("%d", &columns2);

    if (columns1 != rows2) {
        printf("Matrix multiplication not possible!\n");
        return 0;
    }

    printf("Enter the elements of the first matrix:\n");
    for (i = 0; i < rows1; i++) {
        for (j = 0; j < columns1; j++) {
            scanf("%d", &mat1[i][j]);
        }
    }

    printf("Enter the elements of the second matrix:\n");
    for (i = 0; i < rows2; i++) {
        for (j = 0; j < columns2; j++) {
            scanf("%d", &mat2[i][j]);
        }
    }

    matrixMultiplication(mat1, mat2, result, rows1, columns1, columns2);

    printf("Resultant matrix after multiplication:\n");
    displayMatrix(result, rows1, columns2);

    return 0;
}
```