

```

#include <stdio.h>

int main() {
    int a[10][10], rows, cols;
    int i, j, rowSum, colSum;

    printf("Enter number of rows and columns: ");
    scanf("%d %d", &rows, &cols);

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

    // Sum of each row
    printf("\nSum of each row:\n");
    for (i = 0; i < rows; i++) {
        rowSum = 0;
        for (j = 0; j < cols; j++) {
            rowSum += a[i][j];
        }
        printf("Row %d sum = %d\n", i + 1, rowSum);
    }

    // Sum of each column
    printf("\nSum of each column:\n");
    for (j = 0; j < cols; j++) {
        colSum = 0;
        for (i = 0; i < rows; i++) {
            colSum += a[i][j];
        }
        printf("Column %d sum = %d\n", j + 1, colSum);
    }

    return 0;
}

```

Output

Enter number of rows and columns: 2

2

Enter elements of the matrix:

1 2 3 4

Sum of each row:

Row 1 sum = 3

Row 2 sum = 7

Sum of each column:

Column 1 sum = 4

Column 2 sum = 6

=== Code Execution Successful ===

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

```
int arr[10], n = 0;
```

```
void create() {
```

```
    int i;
```

```
    printf("Enter number of elements: ");
```

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

```
    printf("Enter elements:\n");
```

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

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

```
}  
}
```

```
void display() {  
    int i;  
    if (n == 0) {  
        printf("Array is empty.\n");  
        return;  
    }  
    printf("Array elements: ");  
    for (i = 0; i < n; i++) {  
        printf("%d ", arr[i]);  
    }  
    printf("\n");  
}
```

```
void insert() {  
    int pos, val, i;  
    if (n >= 100) {  
        printf("Array is full!\n");  
        return;  
    }  
    printf("Enter position to insert (1 to %d): ", n + 1);  
    scanf("%d", &pos);  
    if (pos < 1 || pos > n + 1) {  
        printf("Invalid position!\n");  
        return;  
    }  
    printf("Enter value to insert: ");  
    scanf("%d", &val);  
  
    for (i = n; i >= pos; i--) {  
        arr[i] = arr[i - 1];  
    }  
    arr[pos - 1] = val;  
    n++;  
    printf("Inserted %d at position %d\n", val, pos);  
}
```

```
void delete() {  
    int pos, i;  
    if (n == 0) {  
        printf("Array is empty.\n");  
        return;  
    }  
    printf("Enter position to delete (1 to %d): ", n);  
    scanf("%d", &pos);  
    if (pos < 1 || pos > n) {
```

```

        printf("Invalid position!\n");
        return;
    }
    printf("Deleted element: %d\n", arr[pos - 1]);
    for (i = pos - 1; i < n - 1; i++) {
        arr[i] = arr[i + 1];
    }
    n--;
}

void linearSearch() {
    int key, i, found = 0;
    if (n == 0) {
        printf("Array is empty.\n");
        return;
    }
    printf("Enter element to search: ");
    scanf("%d", &key);
    for (i = 0; i < n; i++) {
        if (arr[i] == key) {
            printf("Element %d found at position %d\n", key, i + 1);
            found = 1;
            break;
        }
    }
    if (!found) {
        printf("Element %d not found\n", key);
    }
}

int main() {
    int choice;
    while (1) {
        printf("\nMenu:\n1. Create\n2. Display\n3. Insert\n4. Delete\n5. Linear Search\n6.
Exit\n");
        printf("Enter your choice: ");
        scanf("%d", &choice);

        switch (choice) {
            case 1: create(); break;
            case 2: display(); break;
            case 3: insert(); break;
            case 4: delete(); break;
            case 5: linearSearch(); break;
            case 6: return 0;
            default: printf("Invalid choice! Try again.\n");
        }
    }
}

```

}

Output

Menu:

1. Create
2. Display
3. Insert
4. Delete
5. Linear Search
6. Exit

Enter your choice: 1

Enter number of elements: 4

Enter elements:

10 20 30 40

Menu:

1. Create
2. Display
3. Insert
4. Delete
5. Linear Search
6. Exit

Enter your choice: 2

Array elements: 10 20 30 40

Menu:

1. Create
2. Display
3. Insert
4. Delete
5. Linear Search
6. Exit

Enter your choice: 3

Enter position to insert (1 to 5): 2

Enter value to insert: 15

Inserted 15 at position 2

Menu:

1. Create
2. Display
3. Insert
4. Delete
5. Linear Search
6. Exit

Enter your choice: 4

Enter position to delete (1 to 5): 3

Deleted element: 20

Menu:

1. Create
2. Display
3. Insert
4. Delete
5. Linear Search
6. Exit

Enter your choice: 5

Enter element to search: 20

Element 20 not found

Menu:

1. Create
2. Display
3. Insert
4. Delete
5. Linear Search
6. Exit

Enter your choice: 6

=== Code Execution Successful ===

```
#include <stdio.h>
int main() {
    int a[10][10], trans[10][10];
    int rows, cols, i, j;
```

```

printf("Enter rows and columns: ");
scanf("%d %d", &rows, &cols);

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

for (i = 0; i < rows; i++) {
    for (j = 0; j < cols; j++) {
        trans[j][i] = a[i][j];
    }
}

printf("Transpose of the matrix:\n");
for (i = 0; i < cols; i++) {
    for (j = 0; j < rows; j++) {
        printf("%d ", trans[i][j]);
    }
    printf("\n");
}

return 0;
}

```

Output

```

Enter rows and columns: 2 3
Enter matrix elements:
1 2 3
4 5 6
Transpose of the matrix:
1 4
2 5
3 6

```

=== Code Execution Successful ===


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

```
int main() {
```

```
    int A[2][2], B[2][2], C[2][2];
```

```
    int i, j, k;
```

```
    printf("Enter ele of first 2x2 matrix:\n");
```

```
    for (i = 0; i < 2; i++) {
```

```
        for (j = 0; j < 2; j++) {
```

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

```
        }
```

```
    }
```

```
    printf("Enter ele of second 2x2 matrix:\n");
```

```

for (i = 0; i < 2; i++) {
    for (j = 0; j < 2; j++) {
        scanf("%d", &B[i][j]);
    }
}

// Initialize result matrix C to 0
for (i = 0; i < 2; i++) {
    for (j = 0; j < 2; j++) {
        C[i][j] = 0;
    }
}

for (i = 0; i < 2; i++) {
    for (j = 0; j < 2; j++) {
        for (k = 0; k < 2; k++) {
            C[i][j] += A[i][k] * B[k][j];
        }
    }
}

printf(" matrix:\n");
for (i = 0; i < 2; i++) {
    for (j = 0; j < 2; j++) {
        printf("%d ", C[i][j]);
    }
    printf("\n");
}

return 0;
}

```

Output

```
Enter ele of first 2x2 matrix:
1 2
3 4
Enter ele of second 2x2 matrix:
5 6
7 8
  matrix:
19 22
43 50

=== Code Execution Successful ===
```

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

```
int main() {
```

```

int arr[] = {2, 3, 4, 2, 3, 5, 6, 7};
int n = sizeof(arr) / sizeof(arr[0]);
int i, j, k;

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

    for (j = i + 1; j < n; ) {
        if (arr[i] == arr[j]) {
            arr[j] = arr[n - 1];
            n--;
        } else {
            j++;
        }
    }
}
printf("Unique elements:\n");
for (i = 0; i < n; i++) {
    printf("%d ", arr[i]);
}
printf("\n");

return 0;
}

```

Output

```

Unique elements:
2 3 4 7 6 5

```

=== Code Execution Successful ===

```

#include <stdio.h>

```

```

void rev(int arr[], int n) {

```

```

    int l = 0, r = n - 1;
    while (l < r) {

```

```

        int temp = arr[l];
        arr[l] = arr[r];

```

```
        arr[r] = temp;

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

    rev(arr, n);

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

Output

5 4 3 2 1

=== Code Execution Successful ===

```

#include <stdio.h>

int main()
{
    int array[10], position, a, n;

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

    printf("Enter %d elements\n", n);

    for ( a = 0 ; a < n ; a++ )
        scanf("%d", &array[a]);

    printf("Enter the location where you wish to delete element\n");
    scanf("%d", &position);

    if ( position >= n+1 )
        printf("Deletion not possible.\n");

    else
    {
        for ( a = position - 1 ; a < n - 1 ; a++ )
            array[a] = array[a+1];

        printf("Resultant array is\n");

        for( a = 0 ; a < n - 1 ; a++ )
            printf("%d\n", array[a]);
    }
    return 0;
}

```

Output

Enter number of elements in array

5

Enter 5 elements

10 20 30 40 50

Enter the location where you wish to delete element

3

Resultant array is

10

20

40

50

=== Code Execution Successful ===