

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

void main() {
    int n, i;
    printf("Enter number of elements: ");
    scanf("%d", &n);

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

    int smallest = arr[0], sec= arr[0];

    for(i = 1; i < n; i++) {
        if(arr[i] < smallest) {
            sec = smallest;
            smallest = arr[i];
        } else if(arr[i] < sec && arr[i] != smallest) {
            sec = arr[i];
        }
    }

    printf("Second smallest element: %d\n", sec);
}
*/
```

The screenshot shows a terminal window titled 'C:\Users\BMSCECSE-L3-' with the following output:

```
Enter number of elements: 5
Enter elements:
10
4
8
13
3
Second smallest element: 4

Process returned 0 (0x0)   execution time : 10.2
26 s
Press any key to continue.
```

The terminal window has a standard Windows-style title bar and close/minimize/maximize buttons. The command prompt is at the top, followed by the user's input '5', the program's processing steps (elements 10, 4, 8, 13, 3), the result 'Second smallest element: 4', and finally the system message at the bottom.

```
#include <stdio.h>

int main() {
    int n, i, j, sum = 0;

    printf("Enter order of square matrix: ");
    scanf("%d", &n);

    int matrix[n][n];

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

    for(i = 0; i < n; i++) {
        sum += matrix[i][i];
    }

    printf("Sum of left diagonal: %d\n", sum);
    return 0;
}
```

```
C:\Users\BMSCECSE-L3- X + - □ ×
Enter order of square matrix: 3
Enter matrix elements:
1
0
4
5
2
3
2
3
6
Sum of left diagonal: 9
Process returned 0 (0x0) execution time : 13.0
75 s
Press any key to continue.
```

The image shows a Windows desktop environment with two windows open. On the left is a code editor window titled "array.c" with the following C code:

```
Start here X array.c X
2
3 int main() {
4     int rows, cols, i, j;
5
6     printf("Enter number of rows and columns: ");
7     scanf("%d %d", &rows, &cols);
8
9     int matrix[rows][cols];
10
11    printf("Enter matrix elements:\n");
12    for(i = 0; i < rows; i++) {
13        for(j = 0; j < cols; j++) {
14            scanf("%d", &matrix[i][j]);
15        }
16    }
17
18    for(i = 0; i < rows; i++) {
19        int rowSum = 0;
20        for(j = 0; j < cols; j++) {
21            rowSum += matrix[i][j];
22        }
23        printf("Sum of row %d: %d\n", i + 1, rowSum);
24    }
25
26    for(j = 0; j < cols; j++) {
27        int colSum = 0;
28        for(i = 0; i < rows; i++) {
29            colSum += matrix[i][j];
30        }
31        printf("Sum of column %d: %d\n", j + 1, colSum);
32    }
33
34    return 0;
35
36 }
```

To the right is a terminal window titled "C:\Users\BMSCECSE-L3-". The terminal output is as follows:

```
C:\Users\BMSCECSE-L3- X + ~ - x
Enter number of rows and columns: 2
2
Enter matrix elements:
2
3
6
0
Sum of row 1: 5
Sum of row 2: 6
Sum of column 1: 8
Sum of column 2: 3

Process returned 0 (0x0) execution time : 15.8
14 s
Press any key to continue.
```

here X array.c X

```
1 #include <stdio.h>
2
3 int main() {
4     int n, i, j, count = 0;
5
6     printf("Enter number of elements: ");
7     scanf("%d", &n);
8
9     int arr[n];
10    int visited[n];
11
12    printf("Enter elements:\n");
13    for(i = 0; i < n; i++) {
14        scanf("%d", &arr[i]);
15        visited[i] = 0;
16    }
17
18    for(i = 0; i < n; i++) {
19        if(visited[i] == 1)
20            continue;
21
22        int duplicate = 0;
23        for(j = i + 1; j < n; j++) {
24            if(arr[i] == arr[j]) {
25                visited[j] = 1;
26                duplicate = 1;
27            }
28        }
29
30        if(duplicate)
31            count++;
32    }
33
34    printf("Total number of duplicate elements: %d\n", count);
35    return 0;
}
```

C:\Users\BMSCECSE-L3-35\DC X

```
Enter number of elements: 5
Enter elements:
2
3
3
5
6
Total number of duplicate elements: 1
Process returned 0 (0x0) execution time : 11.898 s
Press any key to continue.
```