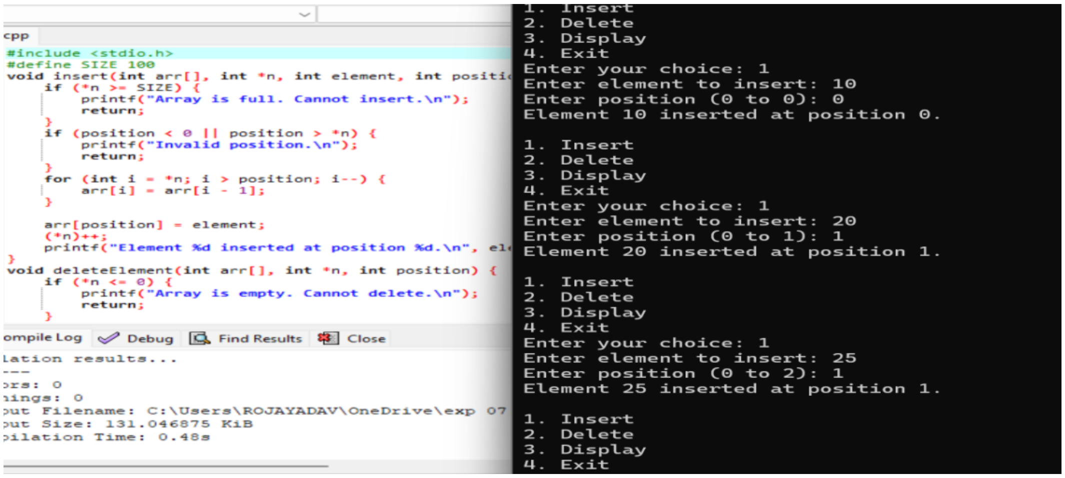
EXP 07 INSERTION, DELETION AND DISPLAY

Sample Code

```
#include <stdio.h>
#define SIZE 100
void insert(int arr[], int *n, int element, int position) {
    if (*n >= SIZE) {
        printf("Array is full. Cannot insert.\n");
        return;
    }
    if (position < 0 || position > *n) {
        printf("Invalid position.\n");
        return;
    }
    for (int i = *n; i > position; i--) {
        arr[i] = arr[i - 1];
    }
    arr[position] = element;
    (*n)++;
    printf("Element %d inserted at position %d.\n", element, position);
}
void deleteElement(int arr[], int *n, int position) {
    if (*n <= 0) {
        printf("Array is empty. Cannot delete.\n");
        return;
    }
    if (position < 0 || position >= *n) {
        printf("Invalid position.\n");
        return;
    }

    int deleted = arr[position];
    for (int i = position; i < *n - 1; i++) {
        arr[i] = arr[i + 1];
    }
    (*n)--;
    printf("Element %d deleted from position %d.\n", deleted, position);
}
void display(int arr[], int n) {
    if (n == 0) {
        printf("Array is empty.\n");
        return;
    }
    printf("Array elements: ");
    for (int i = 0; i < n; i++) {
        printf("%d ", arr[i]);
    }
    printf("\n");
}
int main() {
    int arr[SIZE];
    int n = 0;
    int choice, element, position;
    while (1) {
        printf("\n1. Insert\n2. Delete\n3. Display\n4. Exit\n");
        printf("Enter your choice: ");
        if (scanf("%d", &choice) != 1) {
            printf("Invalid input. Exiting.\n");
            break;
        }
        switch (choice) {
            case 1:
                printf("Enter element to insert: ");
                scanf("%d", &element);
                printf("Enter position (0 to %d): ", n);
                scanf("%d", &position);
                insert(arr, &n, element, position);
                break;
            case 2:
                printf("Enter position to delete (0 to %d): ", n - 1);
                scanf("%d", &position);
                deleteElement(arr, &n, position);
                break;
            case 3:
                display(arr, n);
                break;
            case 4:
                printf("Exiting program.\n");
                return 0;
            default:
                printf("Invalid choice. Please enter 1–4.\n");
        }
    }
    return 0;
}
```

Output



```
#include <stdio.h>
#define SIZE 100
void insert(int arr[], int *n, int element, int positi
    if (*n >= SIZE) {
        printf("Array is full. Cannot insert.\n");
        return;
    }
    if (position < 0 || position > *n) {
        printf("Invalid position.\n");
        return;
    }
    for (int i = *n; i > position; i--) {
        arr[i] = arr[i - 1];
    }

    arr[position] = element;
    (*n)++;
    printf("Element %d inserted at position %d.\n", el
}
void deleteElement(int arr[], int *n, int position) {
    if (*n <= 0) {
        printf("Array is empty. Cannot delete.\n");
        return;
    }
```

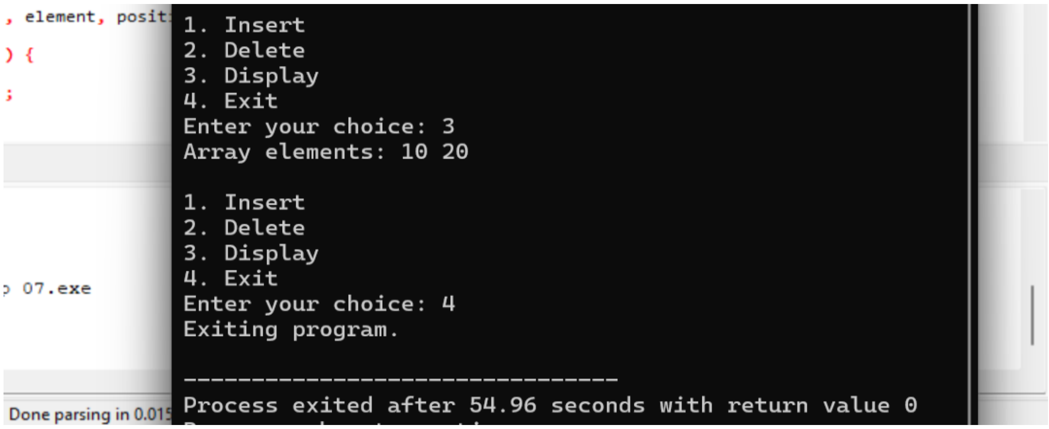
```
lation results...
---
ors: 0
nings: 0
put Filename: C:\Users\ROJAYADAV\OneDrive\exp 07
put Size: 131.046875 KiB
pilation Time: 0.48s
```

```
1. Insert
2. Delete
3. Display
4. Exit
Enter your choice: 1
Enter element to insert: 10
Enter position (0 to 0): 0
Element 10 inserted at position 0.

1. Insert
2. Delete
3. Display
4. Exit
Enter your choice: 1
Enter element to insert: 20
Enter position (0 to 1): 1
Element 20 inserted at position 1.

1. Insert
2. Delete
3. Display
4. Exit
Enter your choice: 1
Enter element to insert: 25
Enter position (0 to 2): 1
Element 25 inserted at position 1.

1. Insert
2. Delete
3. Display
4. Exit
```



```
1. Insert
2. Delete
3. Display
4. Exit
Enter your choice: 3
Array elements: 10 20

1. Insert
2. Delete
3. Display
4. Exit
Enter your choice: 4
Exiting program.

--------------------------------
Process exited after 54.96 seconds with return value 0
```