1. Implement the following array operations on one-dimensional array

a. Insert

b. Delete

c. Display

d. Search

e. Add two arrays

```
#include <stdio.h>
#define MAX 100   // maximum size of array
int arr[MAX], n = 0;  // array and current size

// Function to insert element at the end
void insert(int val) {
    if (n == MAX) {
        printf("\nArray is full!\n");
        return;
    }
    arr[n] = val;
    n++;
```

```
    printf("\nInserted %d at the end.\n", val);
}

// Function to delete element from the end
void delete() {
    if (n == 0) {
        printf("\nArray is empty!\n");
        return;
    }
    printf("\nDeleted %d from the end.\n", arr[n-1]);
    n--;
}

// Function to display array elements
void display() {
    if (n == 0) {
        printf("\nArray is empty!\n");
        return;
    }
```

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

// Function to search element
void search(int val) {
    for (int i = 0; i < n; i++) {
        if (arr[i] == val) {
            printf("\nElement %d found at position %d.\n", val, i);
            return;
        }
    }
    printf("\nElement %d not found!\n", val);
}

// Function to add two arrays
```

```
void addArrays(int a[], int b[], int size) {
    int c[MAX];
    printf("\nSum of arrays: ");
    for (int i = 0; i < size; i++) {
        c[i] = a[i] + b[i];
        printf("%d ", c[i]);
    }
    printf("\n");
}

int main() {
    int choice, val, size;

    while (1) {
        printf("\n--- Array Operations ---\n");
        printf("1. Insert at end\n");
        printf("2. Delete from end\n");
        printf("3. Display\n");
        printf("4. Search\n");
```

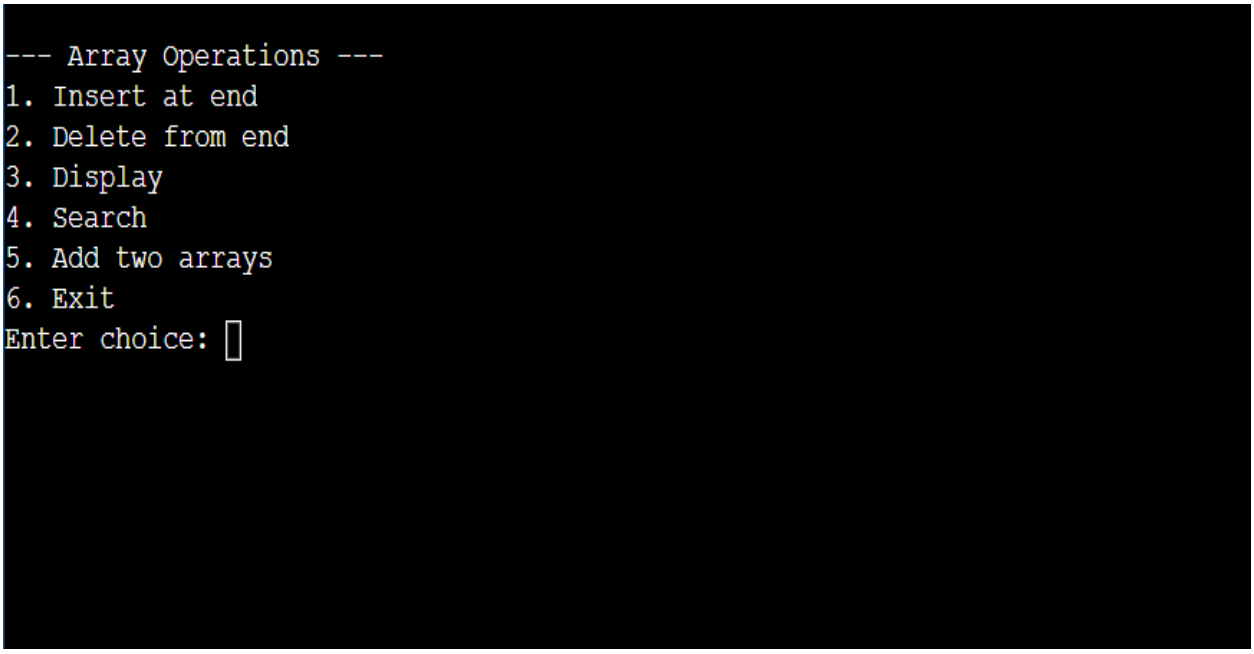
```
printf("5. Add two arrays\n");
printf("6. Exit\n");
printf("Enter choice: ");
scanf("%d", &choice);

switch (choice) {
case 1:
   printf("Enter value to insert: ");
   scanf("%d", &val);
   insert(val);
   break;
case 2:
   delete();
   break;
case 3:
   display();
   break;
```

```
case 4:
    printf("Enter value to search: ");
    scanf("%d", &val);
    search(val);
    break;
 case 5:
    printf("Enter size of arrays: ");
    scanf("%d", &size);
    if (size > MAX) {
       printf("Size too large!\n");
       break;
    }
    int a[MAX], b[MAX];
    printf("Enter elements of first array:\n");
    for (int i = 0; i < size; i++) scanf("%d", &a[i]);
    printf("Enter elements of second array:\n");
    for (int i = 0; i < size; i++) scanf("%d", &b[i]);
    addArrays(a, b, size);
    break;
```

```
        case 6:
            return 0;
        default:
            printf("Invalid choice!\n");
        }
    }
}
```

OUTPUT



```
--- Array Operations ---
1. Insert at end
2. Delete from end
3. Display
4. Search
5. Add two arrays
6. Exit
Enter choice: 1
Enter value to insert: 5

Inserted 5 at the end.

--- Array Operations ---
1. Insert at end
2. Delete from end
3. Display
4. Search
5. Add two arrays
6. Exit
Enter choice: 1
Enter value to insert: 6

Inserted 6 at the end.
```

```
--- Array Operations ---
1. Insert at end
2. Delete from end
3. Display
4. Search
5. Add two arrays
6. Exit
Enter choice: 3

Array elements: 1 5 6
```

```
--- Array Operations ---
1. Insert at end
2. Delete from end
3. Display
4. Search
5. Add two arrays
6. Exit
Enter choice: 2

Deleted 6 from the end.

--- Array Operations ---
1. Insert at end
2. Delete from end
3. Display
4. Search
5. Add two arrays
6. Exit
Enter choice: 3

Array elements: 1 5
```

```
--- Array Operations ---
1. Insert at end
2. Delete from end
3. Display
4. Search
5. Add two arrays
6. Exit
Enter choice: 4
Enter value to search: 5

Element 5 found at position 1.

--- Array Operations ---
1. Insert at end
2. Delete from end
3. Display
4. Search
5. Add two arrays
6. Exit
Enter choice: 4
Enter value to search: 3

Element 3 not found!
```

```
--- Array Operations ---
1. Insert at end
2. Delete from end
3. Display
4. Search
5. Add two arrays
6. Exit
Enter choice: 5
Enter size of arrays: 3
Enter elements of first array:
1
3
5
Enter elements of second array:
1
6
3

Sum of arrays: 2 9 8
```