**7. Write a C program to implement Array operations such as Insert, Delete and Display**

#include <stdio.h>

int main() {

int arr[100], n = 0, choice, pos, val;

while(1) {

printf("\nArray Operations:\n");

printf("1. Insert\t2. Delete\t3. Display\t4. Exit\n");

printf("Enter choice: ");

scanf("%d", &choice);

switch(choice) {

case 1: // Insert

if(n >= 100) {

printf("Array full! Cannot insert.\n");

break;

}

printf("Enter position (0 to %d): ", n);

scanf("%d", &pos);

if(pos < 0 || pos > n) {

printf("Invalid position!\n");

break;

}

printf("Enter value: ");

scanf("%d", &val);

// Shift elements to right

for(int i = n; i > pos; i--)

arr[i] = arr[i-1];

arr[pos] = val;

n++;

printf("Inserted successfully.\n");

break;

case 2: // Delete

if(n == 0) {

printf("Array empty! Cannot delete.\n");

break;

}

printf("Enter position (0 to %d): ", n-1);

scanf("%d", &pos);

if(pos < 0 || pos >= n) {

printf("Invalid position!\n");

break;

}

printf("Deleted element: %d\n", arr[pos]);

// Shift elements to left

for(int i = pos; i < n-1; i++)

arr[i] = arr[i+1];

n--;

break;

case 3: // Display

if(n == 0) {

printf("Array is empty.\n");

} else {

printf("Array elements: ");

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

printf("%d ", arr[i]);

printf("\nSize: %d\n", n);

}

break;

case 4: // Exit

printf("Exiting program.\n");

return 0;

default:

printf("Invalid choice! Try again.\n");

}

}

}