Task-1 Shift Left k Cells

```
#include <stdio.h>
void shiftLeft(int source[], int size, int k) {
 for (int i = 0; i < size - k; i++) {
 source[i] = source[i + k];
 }
 for (int i = size - k; i < size; i++) {
 source[i] = 0;
 }
}
int main() {
 int source[] = {10, 20, 30, 40, 50, 60};
 int k = 3;
 int size = sizeof(source) / sizeof(source[0]);
 printf("Natural Array: [ ");

 for (int i = 0; i < size; i++) {
 printf("%d \n", source[i]);
 }
 printf("]\n");
shiftLeft(source, size, k);
printf("After shifting left by %d positions: [ ", k);
 for (int i = 0; i < size; i++) {
 printf("%d ", source[i]);
  }
 printf("]\n");
 return 0;
}
```

Task-2 Shift Right k Cells

```
#include <stdio.h>
void shiftRight(int source[], int size, int k) {
 for (int i = size - 1; i >= k; i--) {
 source[i] = source[i - k];
 }
 for (int i = 0; i < k; i++) {
 source[i] = 0;
 }
}
```

```
int main() {
 int source[] = {10, 20, 30, 40, 50, 60};
 int k = 3;
 int size = sizeof(source) / sizeof(source[0]);
 printf("Original array: [ ");
 for (int i = 0; i < size; i++) {
 printf("%d ", source[i]);
 }
 printf("]\n");
 shiftRight (source, size, k);
 printf("After shifting Right by %d positions: [ ", k);
 for (int i = 0; i < size; i++) {
 printf("%d ", source[i]);
 }
 printf("]\n");
 return 0;
}
```

Task-3 Rotate Left k cells

```
#include <stdio.h>
void rotateLeft(int source[], int size, int k) {
 int temp[k];
 for (int i = 0; i < k; i++) {
 temp[i] = source[i];
 }
 for (int i = 0; i < size - k; i++) {
 source[i] = source[i + k];
 }
 for (int i = size - k, j = 0; i < size; i++, j++) {
 source[i] = temp[j];
 }
}
int main() {
 int source[] = {10, 20, 30, 40, 50, 60};
 int k = 3;
 int size = sizeof(source) / sizeof(source[0]);
 printf("Original array: [ ");
 for (int i = 0; i < size; i++) {
 printf("%d ", source[i]);
```

```
 };
 printf("]\n");
 rotateLeft(source, size, k);
 printf("After rotating left by %d positions: [ ", k);
 for (int i = 0; i < size; i++) {
 printf("%d ", source[i]);
 }
 printf("]\n");
  return 0;
}
```

Task-4 Rotate Right K Sell

```
#include <stdio.h>
void rotateRight(int source[], int size, int k) {
 int temp[k];
 for (int i = size - k, j = 0; i < size; i++, j++) {
 temp[j] = source[i];
 }
 for (int i = size - 1; i >= k; i--) {
 source[i] = source[i - k];
 }
 for (int i = 0; i < k; i++) {
 source[i] = temp[i];
 }
}
int main() {
 int source[] = {10, 20, 30, 40, 50, 60};
 int k = 3;
 int size = sizeof(source) / sizeof(source[0]);
 printf("Original array: [ ");
 for (int i = 0; i < size; i++) {
 printf("%d ", source[i]);
 }
 printf("]\n");
  rotateRight(source, size, k);
 printf("After rotating right by %d positions: [ ", k);
 for (int i = 0; i < size; i++) {
 printf("%d ", source[i]);
 }
 printf("]\n");
 return 0;
```

```
}
```

Task-5 Remove an element from an array

```
#include <stdio.h>
void removeElement(int source[], int size, int idx) {
 if (idx < 0 || idx >= size) {
 printf("Invalid index. Element not removed.\n");
 return;
 }
 for (int i = idx; i < size - 1; i++) {
 source[i] = source[i + 1];
 }
 source[size - 1] = 0;}
 int main() {
 int source[] = {10, 20, 30, 40, 50, 0, 0};
 int size = 7;
 int idx = 2;
 printf("Original array: [ ");
 for (int i = 0; i < size; i++) {
        printf("%d ", source[i]);
 }
 printf("]\n");
 removeElement(source, size, idx);
 printf("After removing the element at index %d: [ ", idx);
 for (int i = 0; i < size; i++) {
 printf("%d ", source[i]);
 }
 printf("]\n");
 return 0;
}
```

Task-6 Remove all occurrences of a particular element from an array

```
#include <stdio.h>
void removeAll(int source[], int size, int element) {
 int newIndex = 0;
 for (int i = 0; i < size; i++) {
 if (source[i] != element) {
 source[newIndex] = source[i];
```

```
 newIndex++;
 }
 }
 for (int i = newIndex; i < size; i++) {
 source[i] = 0;
 }
}
int main() {
 int source[] = {10, 2, 30, 2, 50, 2, 2, 0, 0};
 int size = 9;
 int element = 2;
 printf("Original array: [ ");
 for (int i = 0; i < size; i++) {
        printf("%d ", source[i]);
 }
 printf("]\n");
 removeAll(source, size, element);
 printf("After removing all occurrences of %d: [ ", element);
 for (int i = 0; i < size; i++) {
 printf("%d ", source[i]);
 }
 printf("]\n");
 return 0;
}
```