```
#include <stdio.h>
  int source[] = {10, 20, 30, 40, 50, 60};
  int n=6;
  void shiftLeft(int source[], int k) {
  for (int j = 0; j < k; j++) {
    for (int i = 0; i < n - 1; i++) {
       source[i] = source[i + 1];
    } source[n - 1] = 0;
  }
}
int main() {
  shiftLeft(source, 3);
  printf("[");
  for (int i = 0; i < n; i++) {
     printf("%d ", source[i]);
  }
printf("]");
  return 0;
}
```

```
#include <stdio.h>
  int source[] = {10, 20, 30, 40, 50, 60};
  int n=6;
  void shifRight(int source[], int k) {
  for (int j = 0; j < k; j++) {
    for (int i = n - 1; i > 0; i--) {
       source[i] = source[i - 1];
    } source[0] = 0;
  }
}
int main() {
  shifRight(source, 3);
  printf("[");
  for (int i = 0; i < n; i++) {
     printf("%d ", source[i]);
  }
printf("]");
  return 0;
}
```

```
#include <stdio.h>
  int source[] = {10, 20, 30, 40, 50, 60};
  int n=6;
  void rotateLeft(int source[], int k) {
  for (int j = 0; j < k; j++) {
       int temp=source[0];
    for (int i = 0; i < n-1; i++) {
       source[i] = source[i + 1];
     } source[n-1] = temp;
  }
}
int main() {
  rotateLeft(source, 3);
  printf("[");
  for (int i = 0; i < n; i++) {
     printf("%d ", source[i]);
  }
printf("]");
  return 0;
}
```

```
#include <stdio.h>
  int source[] = {10, 20, 30, 40, 50, 60};
  int n=6;
  void rotateRight(int source[], int k) {
  for (int j = 0; j < k; j++) {
       int temp=source[n-1];
    for (int i = n - 1; i > 0; i--) {
       source[i] = source[i - 1];
    } source[0] = temp;
  }
}
int main() {
  rotateRight(source, 3);
  printf("[");
  for (int i = 0; i < n; i++) {
    printf("%d ", source[i]);
  }
printf("]");
  return 0;
}
```

```
#include <stdio.h>
  int source[] = {10, 20, 30, 40, 50, 0, 0};
  int size=7;
  void removeE(int source[], int size, int idx) {
  if (idx >= 0 \&\& idx < size) {
    for (int i = idx; i < size - 1; i++) {
       source[i] = source[i + 1];
    }source[size - 1] = 0;
  }
}
int main() {
  removeE(source,5,2);
  printf("After removal: [ ");
  for (int i = 0; i < size; i++) {
    printf("%d ", source[i]);
  }printf("]");
  return 0;
}
```

```
#include <stdio.h>
  int source[] = {10, 2, 30, 2, 50, 2, 2, 0, 0};
  int size = 9;
  void removeAll(int source[], int size, int element) {
  int newSize = 0;
  for (int i = 0; i < size; i++) {
    if (source[i] != element) {
       source[newSize] = source[i];
       newSize++;
    }
  }
  for (int i = newSize; i < size; i++) {
    source[i] = 0;
  }
}
int main() {
  removeAll(source, 7, 2);
  printf("After removal: [ ");
  for (int i = 0; i < size; i++) {
    printf("%d ", source[i]);
  }
  printf("]");
  return 0;
}
```

```
Task7
#include <stdio.h>
#include <stdbool.h>
bool hasTwoElementsWithSameRepetition(int arr[], int size) {
  int countArray[100]={0};
  for (int i=0;i<size;i++) {
    countArray[arr[i]]++;
  }
  int repetitions[100] = {0};
  int maxRepetition = 0;
  for (int i=0;i<100;i++) {
    if (countArray[i] > 1) {
      repetitions[countArray[i]]++;
      maxRepetition=(countArray[i]>maxRepetition)?countArray[i]:maxRepetition;
    }
  }
  for (int i=0;i<100;i++) {
    if (repetitions[i]>=2) {
      return true;
    }
  }
  return false;
}
int main() {
  int arr1[] = {4, 5, 6, 6, 4, 3, 6, 4};
  bool result1 = hasTwoElementsWithSameRepetition(arr1, sizeof(arr1) / sizeof(arr1[0]));
  printf("Output 1: %s\n", result1 ? "True" : "False");
  int arr2[] = {3, 4, 6, 3, 4, 7, 4, 6, 8, 6, 6};
  bool result2 = hasTwoElementsWithSameRepetition(arr2, sizeof(arr2) / sizeof(arr2[0]));
  printf("Output 2: %s\n", result2 ? "True" : "False");
  return 0;
}
```