main.c

```c
#include <stdio.h>
#define MAX 10
void addFront(int *, int, int *, int *);
void addRear(int *, int, int *, int *);
int delFront(int *, int *, int *);
int delRear(int *, int *, int *);
void display(int *);
int count(int *);
int main() {
  int arr[MAX];
  int front, rear, i, n;
  front = rear = -1;
  for (i = 0; i < MAX; i++)
    arr[i] = 0;
  addRear(arr, 5, &front, &rear);
  addFront(arr, 12, &front, &rear);
  addRear(arr, 11, &front, &rear);
  addFront(arr, 5, &front, &rear);
  addRear(arr, 6, &front, &rear);
  addFront(arr, 8, &front, &rear);
  printf("\nElements in a deque: ");
  display(arr);
  i = delFront(arr, &front, &rear);
  printf("\nremoved item: %d", i);
  printf("\nElements in a deque after deletion: ");
  display(arr);
  addRear(arr, 16, &front, &rear);
  addRear(arr, 7, &front, &rear);
  printf("\nElements in a deque after addition: ");
  display(arr);
  i = delRear(arr, &front, &rear);
  printf("\nremoved item: %d", i);
  printf("\nElements in a deque after deletion: ");
  display(arr);
  n = count(arr);
  printf("\nTotal number of elements in deque: %d", n);
}
void addFront(int *arr, int item, int *pfront, int *prear) {
  int i, k, c;
  if (*pfront == 0 && *prear == MAX - 1) {
    printf("\nDeque is full.\n");
    return;
  }
  if (*pfront == -1) {
    *pfront = *prear = 0;
```

```
    *pfront = *prear = 0;
    arr[*pfront] = item;
    return;
  }
  if (*prear != MAX - 1) {
    c = count(arr);
    k = *prear + 1;
    for (i = 1; i <= c; i++) {
      arr[k] = arr[k - 1];
      k--;
    }
    arr[k] = item;
    *pfront = k;
    (*prear)++;
  } else {
    (*pfront)--;
    arr[*pfront] = item;
  }
}
void addRear(int *arr, int item, int *pfront, int *prear) {
  int i, k;
  if (*pfront == 0 && *prear == MAX - 1) {
    printf("\nDeque is full.\n");
    return;
  }
  if (*pfront == -1) {
    *prear = *pfront = 0;
    arr[*prear] = item;
    return;
  }
  if (*prear == MAX - 1) {
    k = *pfront - 1;
    for (i = *pfront - 1; i < *prear; i++) {
      k = i;
      if (k == MAX - 1)
        arr[k] = 0;
      else
        arr[k] = arr[i + 1];
    }
    (*prear)--;
    (*pfront)--;
  }
  (*prear)++;
  arr[*prear] = item;
}
int delFront(int *arr, int *pfront, int *prear) {
```

```
  }
  (*prear)++;
  arr[*prear] = item;
}
int delFront(int *arr, int *pfront, int *prear) {
  int item;
  if (*pfront == -1) {
    printf("\nDeque is empty.\n");
    return 0;
  }
  item = arr[*pfront];
  arr[*pfront] = 0;
  if (*pfront == *prear)
    *pfront = *prear = -1;
  else
    (*pfront)++;
  return item;
}
int delRear(int *arr, int *pfront, int *prear) {
  int item;
  if (*pfront == -1) {
    printf("\nDeque is empty.\n");
    return 0;
  }
  item = arr[*prear];
  arr[*prear] = 0;
  (*prear)--;
  if (*prear == -1)
    *pfront = -1;
  return item;
}
void display(int *arr) {
  int i;
  printf("\n front:  ");
  for (i = 0; i < MAX; i++)
    printf("  %d", arr[i]);
  printf("  :rear");
}
int count(int *arr) {
  int c = 0, i;
  for (i = 0; i < MAX; i++) {
    if (arr[i] != 0)
      c++;
  }
  return c;
}
```