Data Structures Lab

Experiment 4: DEQUE

Project By:

Mohammed Rabeeh
Roll No: 35
TVE18CS036

Contents

1 Aim

To write a menu driven program to implement Input restricted and Output restricted DEQUE.

2 Problem Description

DEQUE stands for Double Ended Queue. It means that insertion and deletion in this type of queue is possible from both the Front and Rear of the queue. There are mainly two types of DEQUEs

1. **Input Restricted DEQUE:** In this type of DEQUE, the elements can be inserted only from the rear but they can be removed from either front or rear.
2. **Output Restricted DEQUE:** In this type of DEQUE, the elements can be inserted from either the front or rear, but they can be removed only from the front.

3 Algorithm

We will be using these function to implement both Input Restricted and Output Restricted DEQUEs

3.1 Insert Left

1. Read input element.
2. If queue is full, return
3. If left is equal to 0, assign left to size - 1. dq[left] = element
4. Else, if left is equal to -1, dq[++left] = element, right++
5. Else dq[–left] = element

3.2 Insert Right

1. If queue is full, return
2. If right is equal to -1, left++
3. dq[++right] = element

3.3 Pop Left

1. If queue is empty, return
2. left = left + 1
3. if left ¿ right, left = -1, right = -1

3.4 Pop Right

1. If queue is empty, return
2. right = right -1
3. if right ¡ left, right = -1, left = -1

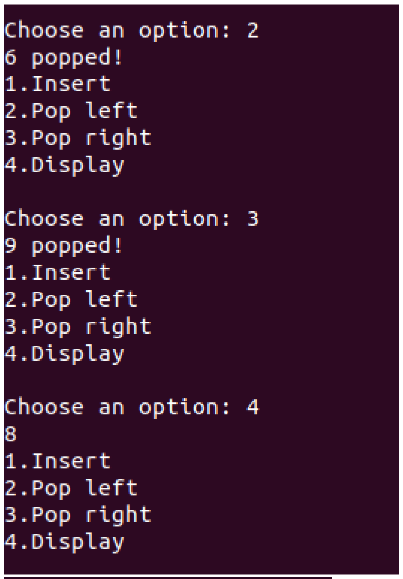
3.5 Display

1. If queue is empty, return
2. If left ¡ right, print elements from left to right
3. if left ¿ right, print elements from left to size - 1 and from 0 index to right

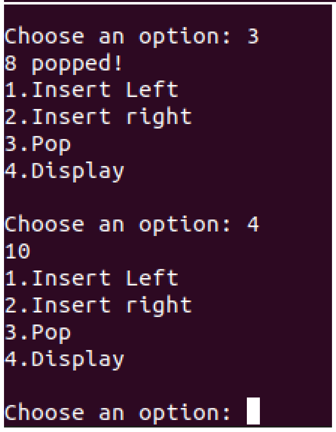
4 Program Code

```c
#include<stdio.h>
#define MAX_SIZE 100
int dq[MAX_SIZE], n, option, i, left = -1, right = -1, size, qsize;

void printvars() {
   printf("\nleft = %d, right = %d, size = %d, qsize = %d\n", left,
       right, size, qsize);
}

int isFull() {
   if(qsize >= size) {
      return 1;
   }
   return 0;
}

void insertRight(int element) {
   if(isFull()) {
      printf("Right overflow\n");
      return ;
   }
   if(right == -1) {
      left++;
   }
   dq[++right] = element;
   qsize++;
   printf("%d successfully inserted!\n", element);
}

void insertLeft(int element) {
   if(isFull()) {
      printf("Queue Full!");
      return ;
   }
   if(left == 0) {
         left = size - 1;
         dq[left] = element;
      return ;
     } else if(left == -1) {
      dq[++left] = element;
      right++;
   } else {
      dq[--left] = element;
   }
   qsize++;
   printf("%d successfully inserted!\n", element);
}

int isEmpty() {
   if(qsize <= 0)
      return 1;
   return 0;
}

void popRight() {
   if(isEmpty()) {
      printf("Nothing to Pop!\n");
      return ;
   }
   printf("%d popped!\n", dq[right]);
   right--;
   if(right < left) {
      right = -1;
      left = -1;
   }
   qsize--;
}

void popLeft() {
   if(isEmpty()) {
      printf("Nothing to Pop!\n");
      return ;
   }
   printf("%d popped!\n", dq[left]);
   left++;
   if(left > right) {
      right = -1;
      left = -1;
   }
   qsize--;
}

void display() {
   int k;
   if(isEmpty()) {
      printf("List empty!\n");
    } else {
      if(left <= right) {
        for(k = left; k <= right; k++) {
          printf("%d ", dq[k]);
        }
      } else {
        for(k = left; k < size; k++) {
          printf("%d ", dq[k]);
        }
        for(k = 0; k <= right; k++) {
          printf("%d ", dq[k]);
        }
      }
      printf("\n");
    }
}

void main() {
    printf("\n\n1.Input Restricted DEQUE\n2.Output Restricted
         DEQUE\n\nChoose an option: ");
    scanf("%d", &option);
    printf("Enter size: ");
    scanf("%d", &size);
    switch(option) {
      case 1:
        while(1) {
          printf("1.Insert\n2.Pop left\n3.Pop
               right\n4.Display\n\nChoose an option: ");
          scanf("%d", &option);
          switch(option) {
            case 1:
              printf("Enter element to insert: ");
              scanf("%d", &n);
              insertRight(n);
              break;
            case 2:
              popLeft();
              break;
            case 3:
                popRight();
                break;
             case 4:
                display();
                break;
             case 5:
                printvars();
                break;
             default:
                printf("Invalid option!\n");

          }
       }
       break;
    case 2:
       while(1) {
          printf("1.Insert Left\n2.Insert
              right\n3.Pop\n4.Display\n\nChoose an option: ");
          scanf("%d", &option);
          switch(option) {
             case 1:
                printf("Enter element to insert: ");
                scanf("%d", &n);
                insertLeft(n);
                break;
             case 2:
                printf("Enter element to insert: ");
                scanf("%d", &n);
                insertRight(n);
                break;
             case 3:
                popLeft();
                break;
             case 4:
                display();
                break;
             case 5:
                printvars();
                break;
             default:
                  printf("Invalid option!\n");

               }
            }
            break;
         default:
            printf("Invalid option!\n");
   }

}
```

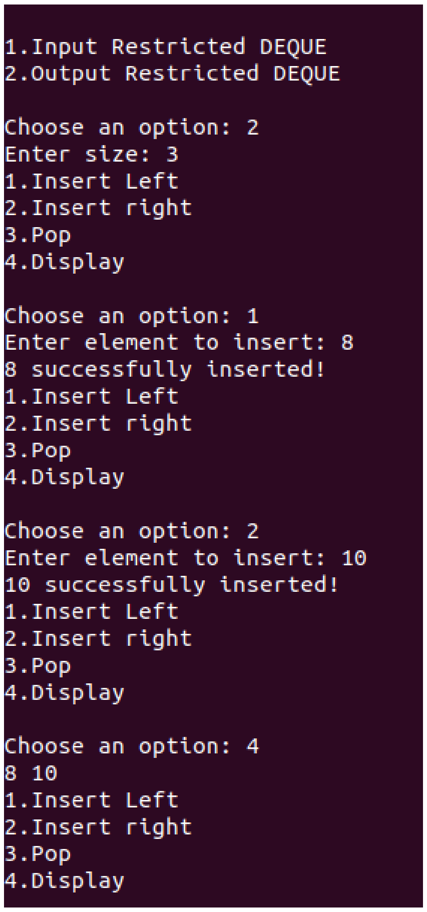
5 Output



```
Choose an option: 2
6 popped!
1.Insert
2.Pop left
3.Pop right
4.Display

Choose an option: 3
9 popped!
1.Insert
2.Pop left
3.Pop right
4.Display

Choose an option: 4
8
1.Insert
2.Pop left
3.Pop right
4.Display
```



```
Choose an option: 3
8 popped!
1.Insert Left
2.Insert right
3.Pop
4.Display

Choose an option: 4
10
1.Insert Left
2.Insert right
3.Pop
4.Display

Choose an option: 
```



```
1.Input Restricted DEQUE
2.Output Restricted DEQUE

Choose an option: 2
Enter size: 3
1.Insert Left
2.Insert right
3.Pop
4.Display

Choose an option: 1
Enter element to insert: 8
8 successfully inserted!
1.Insert Left
2.Insert right
3.Pop
4.Display

Choose an option: 2
Enter element to insert: 10
10 successfully inserted!
1.Insert Left
2.Insert right
3.Pop
4.Display

Choose an option: 4
8 10
1.Insert Left
2.Insert right
3.Pop
4.Display
```

6 Result

Both the Input Restricted DEQUE and Output Restricted DEQUE were implemented in the C language and the functions of these DEQUEs were tested and the output was verified.