PROGRAM

```
#include <stdio.h>

#include <stdlib.h>


void enqueue(int item, int Q[], int n, int *rear) {

   if (*rear == n - 1) {

      printf("Queue is full.\n");

   } else {

      *rear = *rear + 1;

      Q[*rear] = item;

      printf("Element enqueued\n");

   }

}


void dequeue(int *item, int Q[], int *front, int rear) {

   if (*front == rear) {

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

   } else {

      *front = *front + 1;

      *item = Q[*front];

      printf("Element dequeued: %d\n", *item);

   }

}


void display(int Q[], int front, int rear) {

   if (front == rear) {

      printf("Queue is empty\n");

   } else {
        printf("Queue elements: ");
        for (int i = front + 1; i <= rear; i++) {
            printf("%d\t", Q[i]);
        }
        printf("\n");
    }
}

int main() {
    int n, Q[100], item, choice = 0;
    printf("Enter the size of the queue:\n");
    scanf("%d", &n);
    int front = -1; // Initial front position
    int rear = -1;  // Initial rear position

    while (choice != 4) {
        printf("Enter operations:\n 1.enqueue\t2.dequeue\t3.display\t4.exit\n");
        scanf("%d", &choice);

        if (choice == 1) {
            printf("Enter the item to enqueue\n");
            scanf("%d", &item);
            enqueue(item, Q, n, &rear);
        } else if (choice == 2) {
            dequeue(&item, Q, &front, rear);
        } else if (choice == 3) {
            display(Q, front, rear);
        } else if (choice == 4) {
            printf("Exiting...\n");
        } else {
            printf("Invalid choice\n");
        }
    }

    return 0;
}
```

OUTPUT:

```
Enter the size of the queue:
4
Enter operations:
1.enqueue     2.dequeue     3.display        4.exit
1
Enter the item to enqueue
12
Element enqueued
Enter operations:
1.enqueue     2.dequeue     3.display        4.exit
1
Enter the item to enqueue
13
Element enqueued
Enter operations:
1.enqueue     2.dequeue     3.display        4.exit
1
```

```
Enter the item to enqueue

14

Element enqueued

Enter operations:

1.enqueue     2.dequeue     3.display       4.exit

1

Enter the item to enqueue

15

Element enqueued

Enter operations:

1.enqueue     2.dequeue     3.display       4.exit

3

Queue elements: 12 13       14       15

Enter operations:

1.enqueue     2.dequeue     3.display       4.exit

2

Element dequeued: 12

Enter operations:

1.enqueue     2.dequeue     3.display       4.exit

3

Queue elements: 13 14       15

Enter operations:

1.enqueue     2.dequeue     3.display       4.exit

4

Exiting...
```