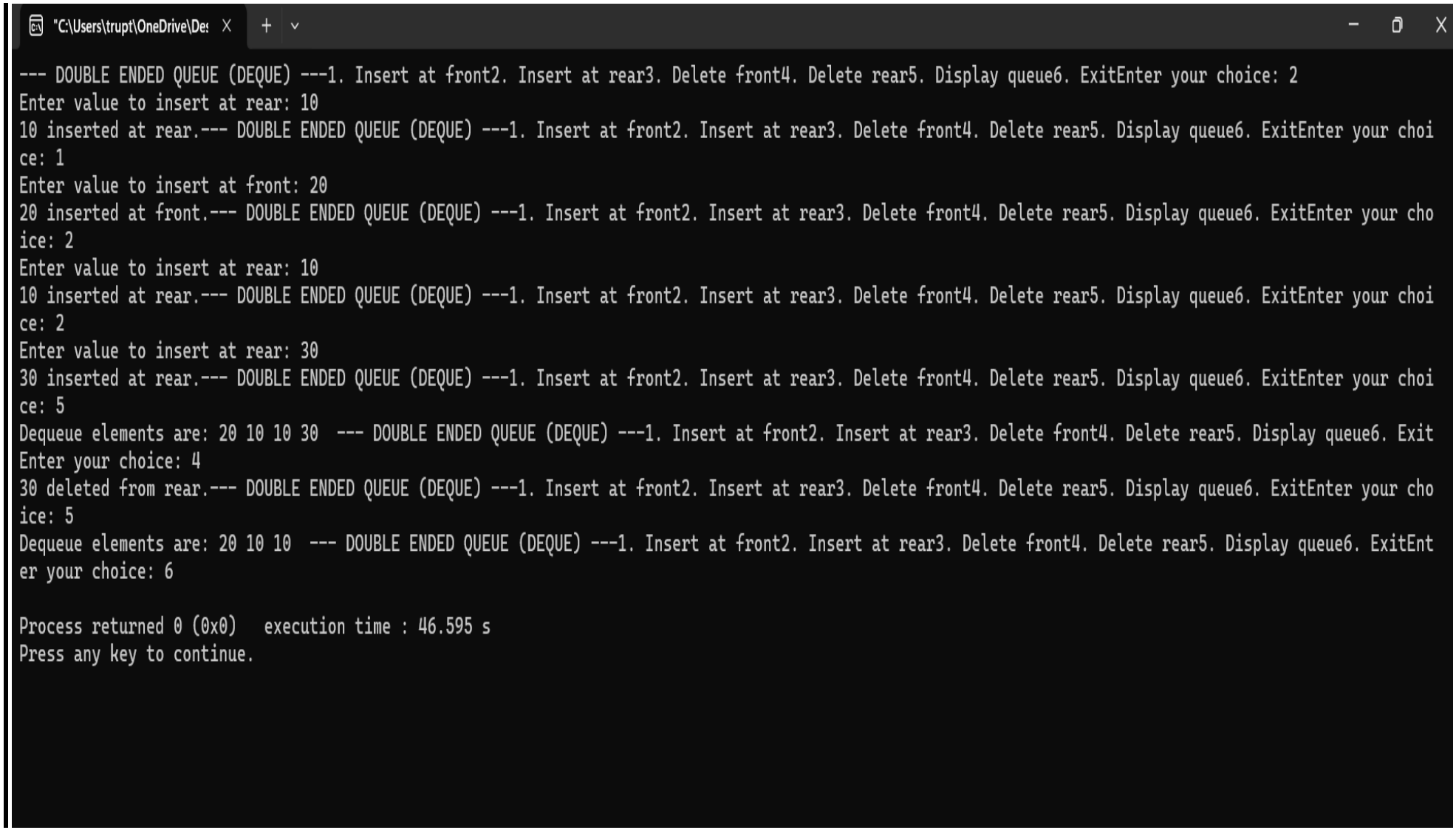
DOUBLE ENDED QUEUE:

```c
#include <stdio.h>
#include <stdlib.h>

#define MAX 5

int data[MAX];
int priority[MAX];
int size = 0;

// Insert an element into the priority queue
void insert(int value, int pr) {
    int i = size - 1;

    if (size == MAX) {
        printf("Queue is full! Cannot insert.\n");
        return;
    }

    // Shift elements to find correct position
    while (i >= 0 && priority[i] > pr) {
        data[i + 1] = data[i];
        priority[i + 1] = priority[i];
        i--;
    }

    data[i + 1] = value;
    priority[i + 1] = pr;
    size++;

    printf("Inserted value %d with priority %d.\n", value, pr);
}

// Delete the element with highest priority
void delete() {
    if (size == 0) {
        printf("Queue is empty! Nothing to delete.\n");
```

```
36          printf("Queue is empty! Nothing to delete.\n");
37          return;
38      }
39
40      printf("Deleted element: %d (Priority: %d)\n", data[0], priority[0]);
41
42      // Shift remaining elements
43      for (int i = 1; i < size; i++) {
44          data[i - 1] = data[i];
45          priority[i - 1] = priority[i];
46      }
47
48      size--;
49  }
50
51  // Display queue
52  void display() {
53      if (size == 0) {
54          printf("Queue is empty.\n");
55          return;
56      }
57
58      printf("\nCurrent Priority Queue:\n");
59      printf("Data\tPriority\n");
60
61      for (int i = 0; i < size; i++) {
62          printf("%d\t%d\n", data[i], priority[i]);
63      }
64  }
65
66  int main() {
67      int choice, value, pr;
68
69      while (1) {
70          printf("\n--- PRIORITY QUEUE USING ARRAY ---\n");
71          printf("1. Insert Element\n");
```

```c
        printf("Enter your choice: ");
        scanf("%d", &choice);

        switch (choice) {
            case 1:
                if (size == MAX) {
                    printf("Queue is full! Cannot insert.\n");
                    break;
                }
                printf("Enter value: ");
                scanf("%d", &value);
                printf("Enter priority (smaller = higher priority): ");
                scanf("%d", &pr);
                insert(value, pr);
                break;

            case 2:
                delete();
                break;

            case 3:
                display();
                break;

            case 4:
                exit(0);

            default:
                printf("Invalid choice! Try again.\n");
        }
    }

    return 0;
}
```


"C:\Users\trupt\OneDrive\De
--- DOUBLE ENDED QUEUE (DEQUE) ---1. Insert at front2. Insert at rear3. Delete front4. Delete rear5. Display queue6. ExitEnter your choice: 2
Enter value to insert at rear: 10
10 inserted at rear.--- DOUBLE ENDED QUEUE (DEQUE) ---1. Insert at front2. Insert at rear3. Delete front4. Delete rear5. Display queue6. ExitEnter your choi
ce: 1
Enter value to insert at front: 20
20 inserted at front.--- DOUBLE ENDED QUEUE (DEQUE) ---1. Insert at front2. Insert at rear3. Delete front4. Delete rear5. Display queue6. ExitEnter your cho
ice: 2
Enter value to insert at rear: 10
10 inserted at rear.--- DOUBLE ENDED QUEUE (DEQUE) ---1. Insert at front2. Insert at rear3. Delete front4. Delete rear5. Display queue6. ExitEnter your choi
ce: 2
Enter value to insert at rear: 30
30 inserted at rear.--- DOUBLE ENDED QUEUE (DEQUE) ---1. Insert at front2. Insert at rear3. Delete front4. Delete rear5. Display queue6. ExitEnter your choi
ce: 5
Dequeue elements are: 20 10 10 30 --- DOUBLE ENDED QUEUE (DEQUE) ---1. Insert at front2. Insert at rear3. Delete front4. Delete rear5. Display queue6. Exit
Enter your choice: 4
30 deleted from rear.--- DOUBLE ENDED QUEUE (DEQUE) ---1. Insert at front2. Insert at rear3. Delete front4. Delete rear5. Display queue6. ExitEnter your cho
ice: 5
Dequeue elements are: 20 10 10 --- DOUBLE ENDED QUEUE (DEQUE) ---1. Insert at front2. Insert at rear3. Delete front4. Delete rear5. Display queue6. ExitEnt
er your choice: 6
Process returned 0 (0x0) execution time : 46.595 s
Press any key to continue.