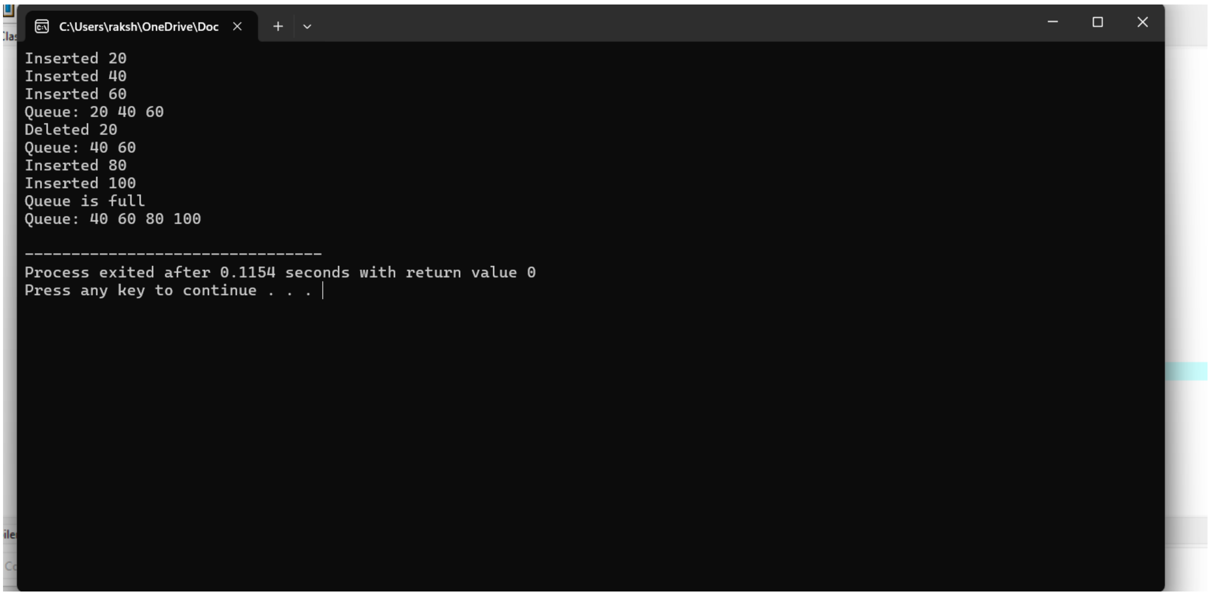
QUEUE IMPLEMENTATION

CODE:

```
#include <stdio.h>
#define SIZE 5
int queue[SIZE];
int front = -1, rear = -1;
void enqueue(int value) {
   if (rear == SIZE - 1)
      printf("Queue is full\n");
   else {
      if (front == -1)
         front = 0;
      rear++;
      queue[rear] = value;
      printf("Inserted %d\n", value);
   }
}
void dequeue() {
   if (front == -1 || front > rear)
      printf("Queue is empty\n");
   else {
      printf("Deleted %d\n", queue[front]);
      front++;
   }
}
void display() {
   if (front == -1 || front > rear)
      printf("Queue is empty\n");
   else {
      printf("Queue: ");
       for (int i = front; i <= rear; i++)
         printf("%d ", queue[i]);
      printf("\n");
   }
}
int main() {
   enqueue(20);
   enqueue(40);
   enqueue(60);
   display();
   dequeue();
   display();
   enqueue(80);
   enqueue(100);
         enqueue(150);
   display();
   return 0;
}
```

OUTPUT:



```
Inserted 20
Inserted 40
Inserted 60
Queue: 20 40 60
Deleted 20
Queue: 40 60
Inserted 80
Inserted 100
Queue is full
Queue: 40 60 80 100

--------------------------------
Process exited after 0.1154 seconds with return value 0
Press any key to continue . . .
```