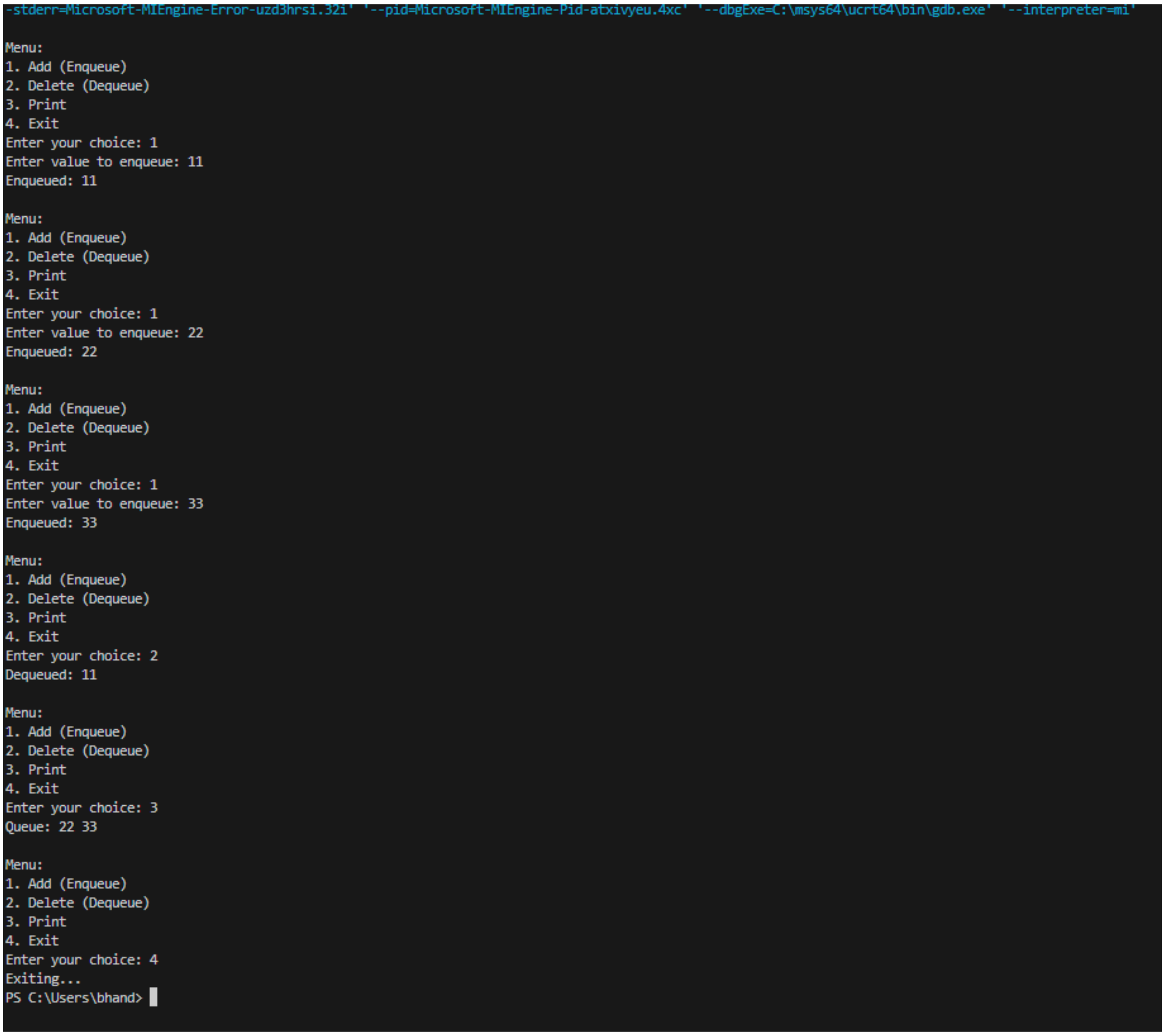
```
#include <stdio.h>
#include <stdlib.h>

// Define the structure for a node in the linked list
struct Node {
    int data;
    struct Node* next;
};

// Define the structure for the queue
struct Queue {
    struct Node *front, *rear;
};

// Function to create a new queue
struct Queue* createQueue() {
    struct Queue* q = (struct Queue*)malloc(sizeof(struct Queue));
    q->front = q->rear = NULL;
    return q;
}

// Function to enqueue an element to the queue
void enqueue(struct Queue* q, int data) {
    struct Node* newNode = (struct Node*)malloc(sizeof(struct Node));
    newNode->data = data;
    newNode->next = NULL;
    if (q->rear == NULL) {
        q->front = q->rear = newNode;
        printf("Enqueued: %d\n", data);
        return;
    }
    q->rear->next = newNode;
    q->rear = newNode;
    printf("Enqueued: %d\n", data);
}

// Function to dequeue an element from the queue
void dequeue(struct Queue* q) {
    if (q->front == NULL) {
        printf("Queue is empty, cannot dequeue.\n");
        return;
    }
    struct Node* temp = q->front;
    printf("Dequeued: %d\n", temp->data);
    q->front = q->front->next;

    if (q->front == NULL)
        q->rear = NULL;

    free(temp);
}

// Function to print the elements of the queue
void printQueue(struct Queue* q) {
    if (q->front == NULL) {
        printf("Queue is empty.\n");
        return;
    }
    struct Node* temp = q->front;
    printf("Queue: ");
    while (temp != NULL) {
        printf("%d ", temp->data);
        temp = temp->next;
    }
    printf("\n");
}

// Main function to provide the menu-driven program
int main() {
    struct Queue* q = createQueue();
    int choice, value;

    while (1) {
        printf("\nMenu:\n");
        printf("1. Add (Enqueue)\n");
        printf("2. Delete (Dequeue)\n");
        printf("3. Print\n");
        printf("4. Exit\n");
        printf("Enter your choice: ");
        scanf("%d", &choice);

        switch (choice) {
            case 1:
                printf("Enter value to enqueue: ");
                scanf("%d", &value);
                enqueue(q, value);
                break;
            case 2:
                dequeue(q);
                break;
            case 3:
                printQueue(q);
                break;
            case 4:
                printf("Exiting...\n");
                exit(0);
            default:
                printf("Invalid choice, try again.\n");
        }
    }

    return 0;
}
```



```
-stderr=Microsoft-MIEngine-Error-uzd3hrsi.32i' '--pid=Microsoft-MIEngine-Pid-atxivyeu.4xc' '--dbgExe=C:\msys64\ucrt64\bin\gdb.exe' '--interpreter=mi'

Menu:
1. Add (Enqueue)
2. Delete (Dequeue)
3. Print
4. Exit
Enter your choice: 1
Enter value to enqueue: 11
Enqueued: 11

Menu:
1. Add (Enqueue)
2. Delete (Dequeue)
3. Print
4. Exit
Enter your choice: 1
Enter value to enqueue: 22
Enqueued: 22

Menu:
1. Add (Enqueue)
2. Delete (Dequeue)
3. Print
4. Exit
Enter your choice: 1
Enter value to enqueue: 33
Enqueued: 33

Menu:
1. Add (Enqueue)
2. Delete (Dequeue)
3. Print
4. Exit
Enter your choice: 2
Dequeued: 11

Menu:
1. Add (Enqueue)
2. Delete (Dequeue)
3. Print
4. Exit
Enter your choice: 3
Queue: 22 33

Menu:
1. Add (Enqueue)
2. Delete (Dequeue)
3. Print
4. Exit
Enter your choice: 4
Exiting...
PS C:\Users\bhand> 
```