Practical 8b

Pra8B Implement a Queue using Linked List and perform the Queue operations: Enqueue , Dequeue and Print using Menu Driver Program such as 1.Add, 2.Delete and 3. Print and 4. Exit.

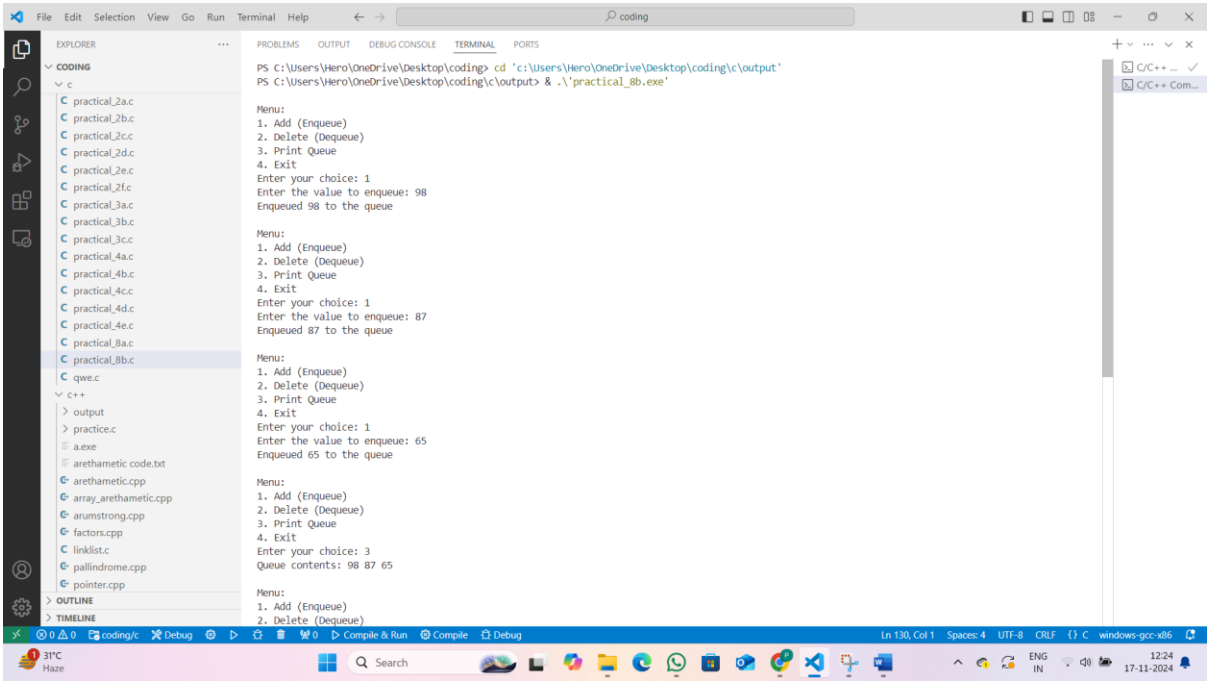
```
#include <stdio.h>
#include <stdlib.h>

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

// Define the Queue structure with pointers to the front and rear nodes
struct Queue {
    struct Node* front;
    struct Node* rear;
};

// Function to initialize the queue
void initializeQueue(struct Queue* q) {
    q->front = NULL;
    q->rear = NULL;
}

// Function to check if the queue is empty
int isEmpty(struct Queue* q) {
    return q->front == NULL;
}

// Function to enqueue (add) an item to the queue
void enqueue(struct Queue* q, int value) {
    struct Node* newNode = (struct Node*)malloc(sizeof(struct Node));
```

```
    if (!newNode) {
        printf("Memory allocation failed.\n");
        return;
    }
    newNode->data = value;
    newNode->next = NULL;

    if (isEmpty(q)) {
        q->front = q->rear = newNode;  // If the queue is empty, both front and rear point to the new node
    } else {
        q->rear->next = newNode;  // Add the new node to the rear
        q->rear = newNode;        // Update the rear pointer
    }

    printf("Enqueued %d to the queue\n", value);
}

// Function to dequeue (remove) an item from the queue
void dequeue(struct Queue* q) {
    if (isEmpty(q)) {
        printf("Queue is empty. Cannot dequeue.\n");
        return;
    }
    struct Node* temp = q->front;
    int removedValue = temp->data;
    q->front = q->front->next;  // Move the front pointer to the next node

    // If the queue becomes empty, update the rear pointer to NULL
    if (q->front == NULL) {
        q->rear = NULL;
    }

    free(temp);  // Free the memory of the removed node
```

```
    printf("Dequeued %d from the queue\n", removedValue);
}

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

// Menu function to drive the program
void menu() {
    struct Queue q;
    initializeQueue(&q);  // Initialize the queue
    int choice, value;

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

```
        switch (choice) {
            case 1:
                // Enqueue operation
                printf("Enter the value to enqueue: ");
                scanf("%d", &value);
                enqueue(&q, value);
                break;

            case 2:
                // Dequeue operation
                dequeue(&q);
                break;

            case 3:
                // Print the queue
                printQueue(&q);
                break;

            case 4:
                // Exit the program
                printf("Exiting the program.\n");
                exit(0);

            default:
                printf("Invalid choice. Please enter a valid option.\n");
        }
    }
}

int main() {
    menu();  // Start the menu-driven program
    return 0;
```

}


File Edit Selection View Go Run Terminal Help
coding
EXPLORER
CODING
c
practical_2a.c
practical_2b.c
practical_2c.c
practical_2d.c
practical_2e.c
practical_2f.c
practical_3a.c
practical_3b.c
practical_3c.c
practical_4a.c
practical_4b.c
practical_4c.c
practical_4d.c
practical_4e.c
practical_8a.c
practical_8b.c
qwe.c
c++
output
practice.c
a.exe
arethametic code.txt
arethametic.cpp
array_arethametic.cpp
arumstrong.cpp
factors.cpp
linklist.c
pallindrome.cpp
pointer.cpp
OUTLINE
TIMELINE
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\Users\Hero\OneDrive\Desktop\coding> cd 'c:\Users\Hero\OneDrive\Desktop\coding\c\output'
PS C:\Users\Hero\OneDrive\Desktop\coding\c\output> & .\'practical_8b.exe'
Menu:
1. Add (Enqueue)
2. Delete (Dequeue)
3. Print Queue
4. Exit
Enter your choice: 1
Enter the value to enqueue: 98
Enqueued 98 to the queue
Menu:
1. Add (Enqueue)
2. Delete (Dequeue)
3. Print Queue
4. Exit
Enter your choice: 1
Enter the value to enqueue: 87
Enqueued 87 to the queue
Menu:
1. Add (Enqueue)
2. Delete (Dequeue)
3. Print Queue
4. Exit
Enter your choice: 1
Enter the value to enqueue: 65
Enqueued 65 to the queue
Menu:
1. Add (Enqueue)
2. Delete (Dequeue)
3. Print Queue
4. Exit
Enter your choice: 3
Queue contents: 98 87 65
Menu:
1. Add (Enqueue)
2. Delete (Dequeue)
C/C++ ...
C/C++ Com...
Compile & Run
Compile
Debug
Ln 130, Col 1
Spaces: 4
UTF-8
CRLF
C
windows-gcc-x86
31°C
Haze
Search
ENG
IN
12:24
17-11-2024