SCHOOL OF ENGINEERING
JNU-DELHI

Name: Sarvesh M. Patil

Roll No: 22/11/EC/030

OPERATING SYSTEM ASSIGNMENT 2

➢ CODE:

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

typedef struct PCB {
    int id;
    char state[20];
    long timestamp;
    struct PCB *next;
} PCB;

// Global pointer to the first PCB in the queue
PCB *head = NULL;

// Function declarations
void addPCB();          // Function to insert a new PCB into the queue
void removePCB();       // Function to remove the PCB at the front of the queue
void showQueue();       // Function to display the contents of the PCB queue

// Entry point of the program
int main() {
    int option;
    int continueProgram = 1;

    do {
        printf("Choose an action:\n");
        printf("1: Add a new PCB\n");
        printf("2: Remove the first PCB\n");
        printf("3: Show the PCB Queue\n");
        printf("Enter your choice: ");
        scanf("%d", &option);

        switch(option) {
            case 1:
                addPCB();
                break;
            case 2:
                removePCB();
                break;
            case 3:
                showQueue();
```

```
            break;
        default:
            printf("Invalid choice! Please try again.\n");
        }

        printf("Would you like to perform another operation? (1 for Yes / 0 for No): ");
        scanf("%d", &continueProgram);
    } while (continueProgram);

    return 0;
}

// Function to add a new PCB to the queue
void addPCB() {
    PCB *newPCB = (PCB *)malloc(sizeof(PCB));

    printf("Enter the PCB ID: ");
    scanf("%d", &newPCB->id);
    printf("Enter the PCB state: ");
    scanf("%s", newPCB->state);
    printf("Enter the PCB timestamp: ");
    scanf("%ld", &newPCB->timestamp);

    newPCB->next = NULL;

    if (head == NULL) {
        head = newPCB;
    } else {
        PCB *current = head;
        while (current->next != NULL) {
            current = current->next;
        }
        current->next = newPCB;
    }
    printf("PCB with ID %d has been added.\n", newPCB->id);
}

// Function to remove the first PCB from the queue
void removePCB() {
    if (head == NULL) {
        printf("The queue is empty, no PCB to remove.\n");
        return;
    }

    PCB *temp = head;
    head = head->next;
```

```
    printf("PCB with ID %d has been removed.\n", temp->id);
    free(temp);
}

// Function to display all the PCBs in the queue
void showQueue() {
    if (head == NULL) {
        printf("The PCB queue is currently empty.\n");
        return;
    }

    PCB *current = head;
    printf("Current PCB Queue:\n");
    while (current != NULL) {
        printf("ID: %d, State: %s, Timestamp: %ld\n", current->id, current->state, current-
>timestamp);
        current = current->next;
    }
}
```

➢ WORKING :

[ADD, REMOVE, DISPLAY]