S.No: 1	Exp. Name: ***Project Module***	Date: 2024-06-13

Aim:

Project Module

Source Code:

hello.c

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

// Define the structure for a job
typedef struct Job {
    int id;
    int priority;
    char description[100];
    struct Job* next;
} Job;

// Define the structure for the job queue
typedef struct JobQueue {
    Job* front;
    int count;
} JobQueue;

// Function to create a new job
Job* createJob(int id, int priority, const char* description) {
    Job* newJob = (Job*)malloc(sizeof(Job));
    newJob->id = id;
    newJob->priority = priority;
    strcpy(newJob->description, description);
    newJob->next = NULL;
    return newJob;
}

// Function to add a job to the queue
void addJob(JobQueue* queue, int priority, const char* description) {
    Job* newJob = createJob(queue->count + 1, priority, description);
    queue->count++;

    // If queue is empty, add job at front
    if (queue->front == NULL) {
        queue->front = newJob;
    } else {
        Job* temp = queue->front;
        Job* prev = NULL;

        // Find correct position based on priority
        while (temp != NULL && temp->priority <= priority) {
            prev = temp;
            temp = temp->next;
        }

        // Insert new job in the queue
        if (prev == NULL) {
            newJob->next = queue->front;
```

```
}

// Function to schedule (display) jobs in priority order
void scheduleJobs(JobQueue* queue) {
    if (queue->front == NULL) {
        printf("No jobs to schedule.\n");
        return;
    }

    printf("Scheduled Jobs:\n");
    Job* temp = queue->front;
    while (temp != NULL) {
        printf("ID: %d, Priority: %d, Description: %s\n", temp->id, temp-
>priority, temp->description);
        temp = temp->next;
    }
}

// Function to process (remove) jobs from the queue
void processJobs(JobQueue* queue) {
    if (queue->front == NULL) {
        printf("No jobs to process.\n");
        return;
    }

    printf("Processing Jobs:\n");
    while (queue->front != NULL) {
        Job* temp = queue->front;
        printf("Processing Job ID: %d, Priority: %d, Description: %s\n", temp-
>id, temp->priority, temp->description);
        queue->front = queue->front->next;
        free(temp);
    }
}

// Main function with job scheduler menu
int main() {
    JobQueue queue = {NULL, 0};
    int choice, priority;
    char description[100];

    while (1) {
        printf("\nJob Scheduler Menu:\n");
        printf("1. Add Job\n");
        printf("2. Schedule Jobs\n");
        printf("3. Process Jobs\n");
        printf("4. Exit\n");
        printf("Enter your choice: ");
        scanf("%d", &choice);
```

```
                break;
            case 2:
                scheduleJobs(&queue);
                break;
            case 3:
                processJobs(&queue);
                break;
            case 4:
                exit(0);
            default:
                printf("Invalid choice. Please try again.\n");
        }
    }

    return 0;
}
```

Execution Results - All test cases have succeeded!

Test Case - 1
User Output
Hello World