PIMPRI CHINCHWAD EDUCATION TRUST's.

PIMPRI CHINCHWAD COLLEGE OF ENGINEERING

(An Autonomous Institute)

S.Y. B. TECH	**Year:** 2024 – 25	**Semester:** I
Name: Abhishek Joshi		**PRN: 123B1B150**
Department: Computer Engineering		**Division:** C (C1)
Course: Data Structures Laboratory		**Course Code:** BCE23PC02
Date:		

Assignment – 10

- **Aim:**

Implement a job scheduling system for a manufacturing plant using a double-ended queue. The system needs to efficiently manage the processing of jobs on various machines throughout the plant. Each job has a Job_priority. The system should support the following operations:

a. Add Job
b. Remove Job
c. Display Job
d. Search Job

- **Source Code :**

```
#include <iostream>
#include <string>
using namespace std;

class Job {
public:
    string name;
    int priority;
    Job* front;
    Job* rear;

    Job(string job_name = "", int p = 0) {
        name = job_name;
        priority = p;
        front = NULL;
        rear = NULL;
    }
};
```

```
class JobQueue {
   Job* head;
public:
   JobQueue() {
      head = NULL;
   }
   void add_Job(string job_name, int p) {
      Job* new_job = new Job(job_name, p);
      if (!head) {
         head = new_job;
         return;
      }
      Job* temp = head;
      Job* prev = NULL;
      while (temp && temp->priority >= p) {
         prev = temp;
         temp = temp->front;
      }
      if (prev == NULL) {
         new_job->front = head;
         head->rear = new_job;
         head = new_job;
      } else {
         new_job->front = temp;
         new_job->rear = prev;
         prev->front = new_job;
         if (temp) temp->rear = new_job;
      }
      cout << "Added Job: " << job_name << " with priority: " << p << endl;
   }

   void remove_job() {
      if (!head) {
         cout << "No jobs to remove." << endl;
         return;
      }
      Job* temp = head;
      head = head->front;
      if (head) {
         head->rear = NULL;
      }

      cout << "Removed Job: " << temp->name << " with priority: " << temp->priority << endl;
```

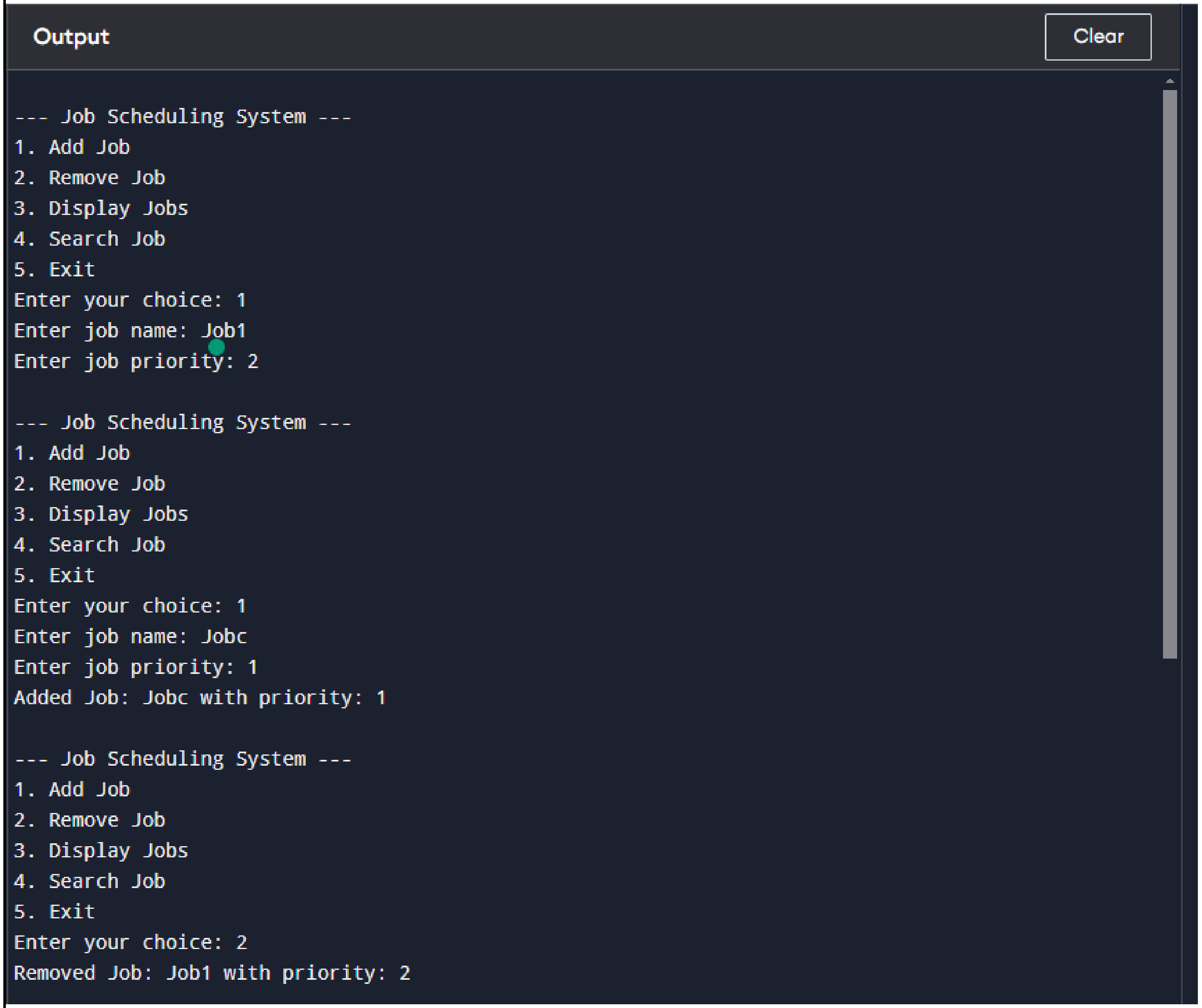
```
            delete temp;
        }

        void searchJob(string job_name) {
            Job* temp = head;
            while (temp) {
                if (temp->name == job_name) {
                    cout << "Job found: " << temp->name << " with priority: " << temp->priority << endl;
                    return;
                }
                temp = temp->front;
            }
            cout << "Job " << job_name << " not found in the queue." << endl;
        }

        void DisplayJob() {
            if (!head) {
                cout << "No jobs in the queue." << endl;
                return;
            }

            Job* temp = head;
            cout << "\nJobs in the queue:" << endl;
            while (temp) {
                cout << "Job: " << temp->name << " with priority: " << temp->priority << endl;
                temp = temp->front;
            }
        }
    };
    int main() {
        JobQueue jq;
        int choice;
        string name;
        int priority;
        do {
            cout << "\n--- Job Scheduling System ---" << endl;
            cout << "1. Add Job" << endl;
            cout << "2. Remove Job" << endl;
            cout << "3. Display Jobs" << endl;
            cout << "4. Search Job" << endl;
            cout << "5. Exit" << endl;
            cout << "Enter your choice: ";
            cin >> choice;
        switch (choice) {
            case 1:
                cout << "Enter job name: ";
                cin >> name;
                cout << "Enter job priority: ";
                cin >> priority;
                jq.add_Job(name, priority);
                break;
            case 2:
                jq.remove_job();
                break;
            case 3:
                jq.DisplayJob();
                break;
            case 4:
                cout << "Enter job name to search: ";
                cin >> name;
                jq.searchJob(name);
                break;
            case 5:
                cout << "Exiting system." << endl;
                break;
            default:
                cout << "Invalid choice! Please try again." << endl;
        }
    } while (choice != 5);
    return 0;
}
```

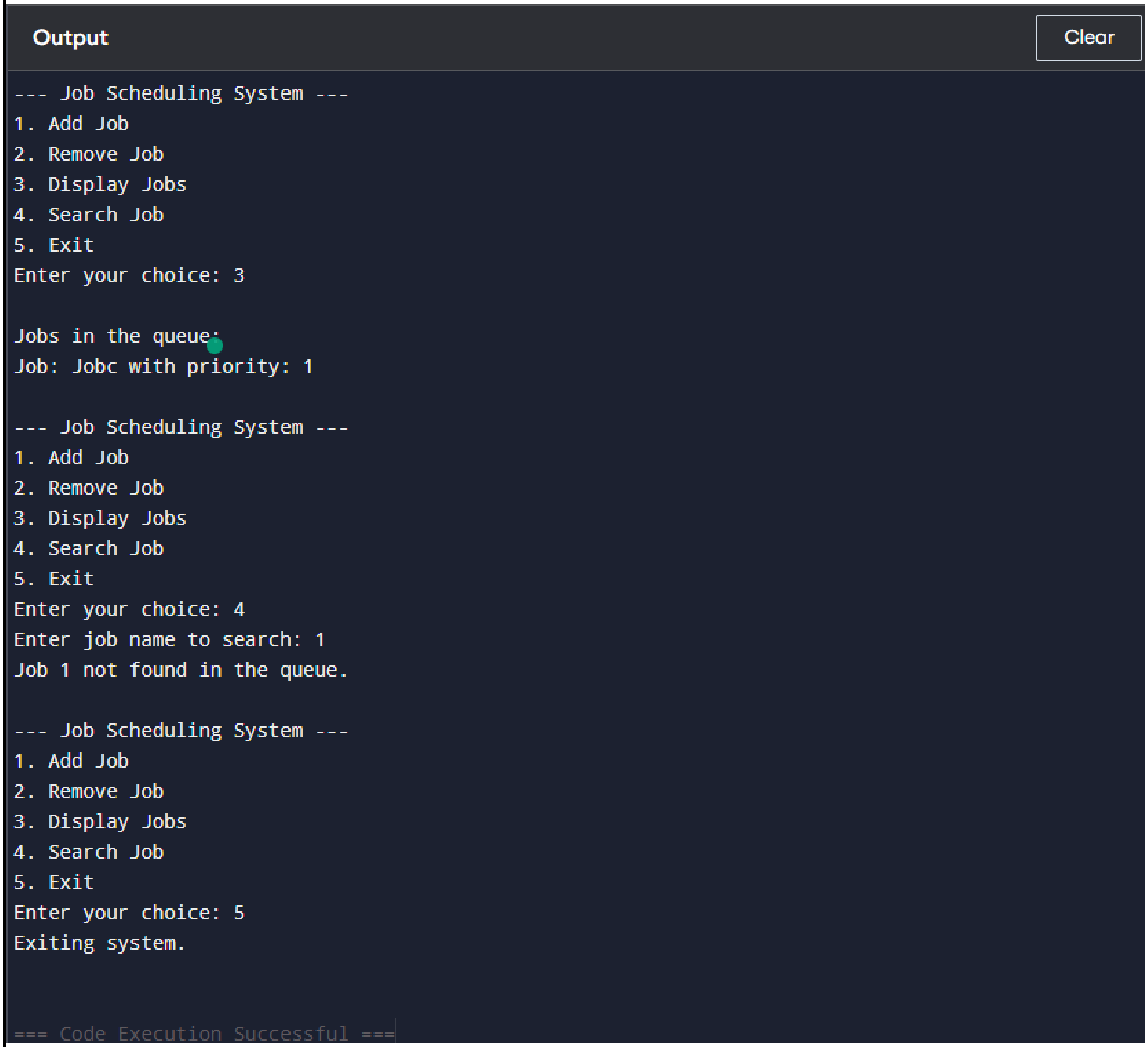
- **Screen Shot of Output :**



```
--- Job Scheduling System ---
1. Add Job
2. Remove Job
3. Display Jobs
4. Search Job
5. Exit
Enter your choice: 1
Enter job name: Job1
Enter job priority: 2

--- Job Scheduling System ---
1. Add Job
2. Remove Job
3. Display Jobs
4. Search Job
5. Exit
Enter your choice: 1
Enter job name: Jobc
Enter job priority: 1
Added Job: Jobc with priority: 1

--- Job Scheduling System ---
1. Add Job
2. Remove Job
3. Display Jobs
4. Search Job
5. Exit
Enter your choice: 2
Removed Job: Job1 with priority: 2
```



```
--- Job Scheduling System ---
1. Add Job
2. Remove Job
3. Display Jobs
4. Search Job
5. Exit
Enter your choice: 3

Jobs in the queue:
Job: Jobc with priority: 1

--- Job Scheduling System ---
1. Add Job
2. Remove Job
3. Display Jobs
4. Search Job
5. Exit
Enter your choice: 4
Enter job name to search: 1
Job 1 not found in the queue.

--- Job Scheduling System ---
1. Add Job
2. Remove Job
3. Display Jobs
4. Search Job
5. Exit
Enter your choice: 5
Exiting system.


=== Code Execution Successful ===
```

- **<u>Conclusion:</u>**
  In this assignment, we implemented a job scheduling system using a double-ended queue to manage job priorities efficiently. The program supports adding, removing, displaying, and searching jobs based on priority, demonstrating essential queue operations.