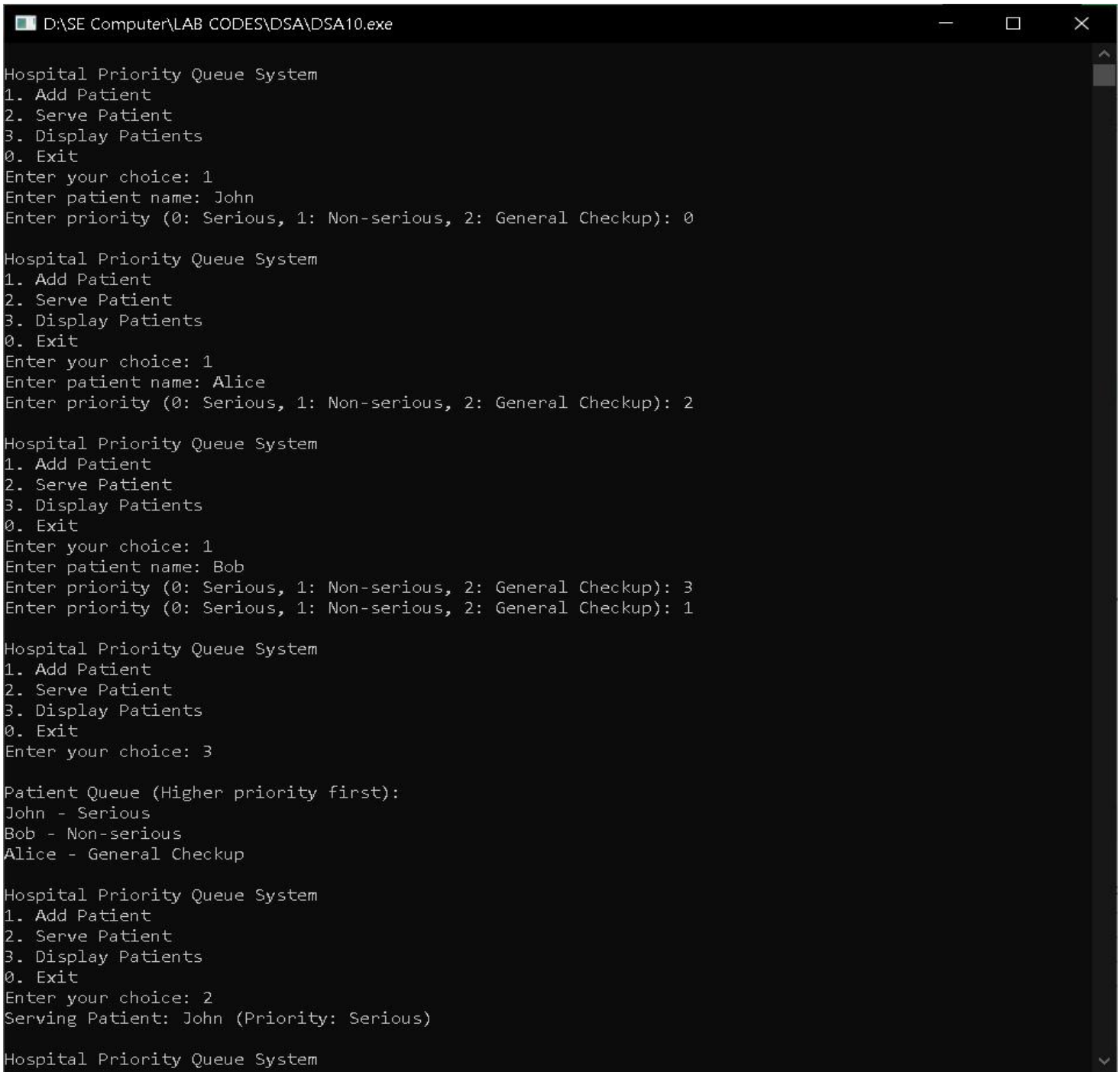
```cpp
#include <iostream>
#include <string>

#define N 20  // Maximum size of the queue

#define SERIOUS 10
#define NONSERIOUS 5
#define CHECKUP 1

using namespace std;

struct Patient
{
    string name;
    int priority;
};

// Priority Queue implementation
class PriorityQueue
{
private:
    Patient Q[N];  // Array to store patient data
    int front, rear;

public:
    PriorityQueue()
        {
        front = rear = -1;
    }

    bool isEmpty()
        {
        return front == -1;
    }

    bool isFull()
        {
        return (rear == N - 1);
    }

    void enqueue(string name, int priority)
        {
        if (isFull())
                {
            cout << "Queue is full! Cannot add more patients.\n";
            return;
        }

        // If queue is empty
        if (isEmpty())
                {
            front = rear = 0;
            Q[rear] = {name, priority};
        }
                else
```

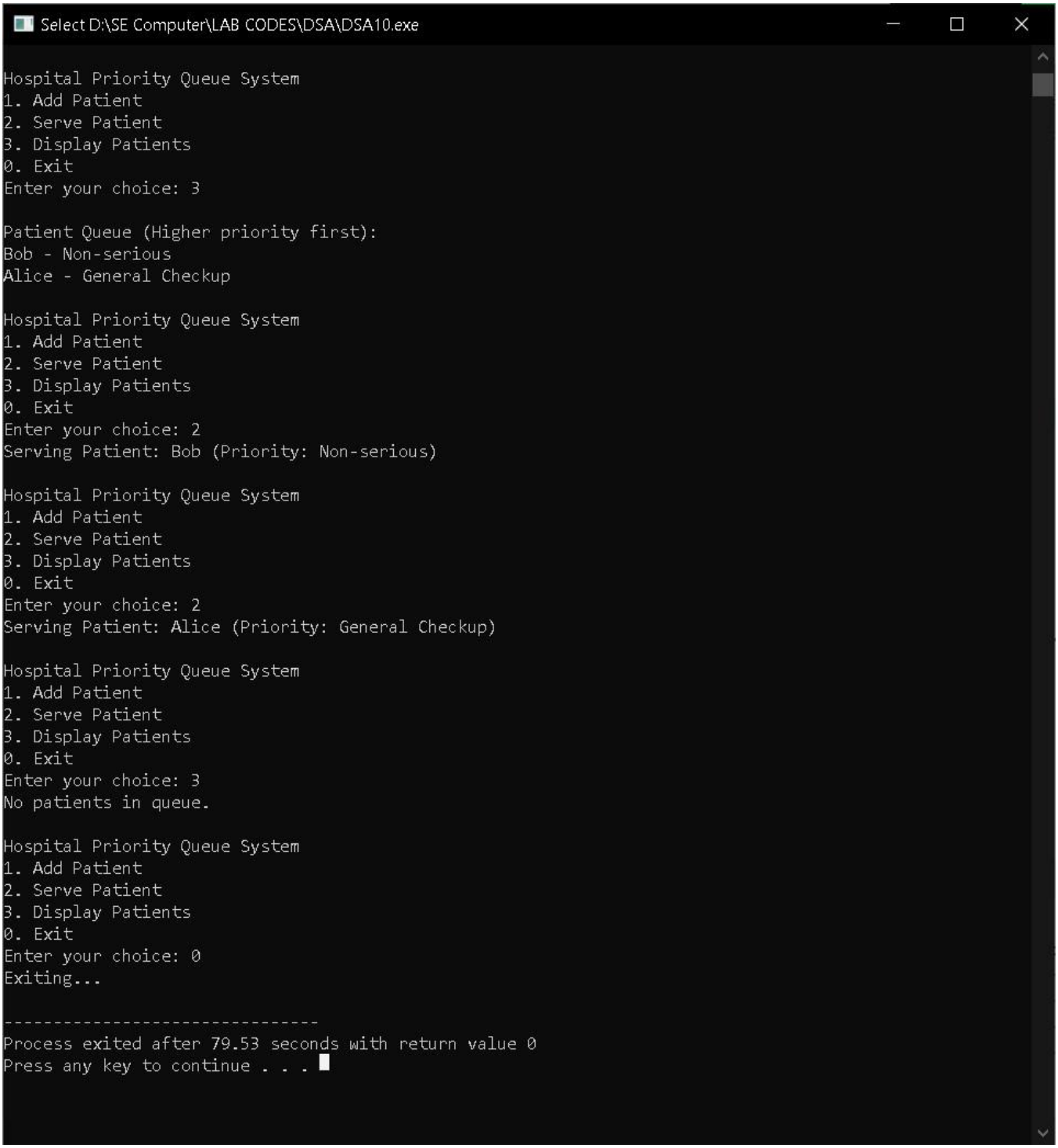
```
56                  {
57              // Insert based on priority
58              int i;
59              for (i = rear; i >= front && Q[i].priority < priority; i--)
60                          {
61                  Q[i + 1] = Q[i];  // Shift elements
62              }
63              Q[i + 1] = {name, priority};
64              rear++;
65          }
66      }
67
68      void dequeue()
69          {
70          if (isEmpty())
71                  {
72              cout << "Queue is empty! No patients to serve.\n";
73              return;
74          }
75
76          cout << "Serving Patient: " << Q[front].name << " (Priority: ";
77          switch (Q[front].priority)
78                  {
79              case SERIOUS: cout << "Serious"; break;
80              case NONSERIOUS: cout << "Non-serious"; break;
81              case CHECKUP: cout << "General Checkup"; break;
82              default: cout << "Unknown"; break;
83          }
84          cout << ")\n";
85
86          // If only one patient in the queue
87          if (front == rear)
88                  {
89              front = rear = -1;
90          }
91                  else
92                  {
93              front++;
94          }
95      }
96
97      void display()
98          {
99          if (isEmpty())
100                 {
101             cout << "No patients in queue.\n";
102             return;
103         }
104         cout << "\nPatient Queue (Higher priority first):\n";
105         for (int i = front; i <= rear; i++)
106                 {
107             cout << Q[i].name << " - ";
108             switch (Q[i].priority)
109                         {
110                 case SERIOUS: cout << "Serious"; break;
111                 case NONSERIOUS: cout << "Non-serious"; break;
112                 case CHECKUP: cout << "General Checkup"; break;
```

```
                default: cout << "Unknown"; break;
            }
            cout << endl;
        }
    }
};

int main()
{
    PriorityQueue pq;
    int choice, priority;
    string name;

    do {
        cout << "\nHospital Priority Queue System\n";
        cout << "1. Add Patient\n";
        cout << "2. Serve Patient\n";
        cout << "3. Display Patients\n";
        cout << "0. Exit\n";
        cout << "Enter your choice: ";
        cin >> choice;

        switch (choice)
                {
            case 1:
                cout << "Enter patient name: ";
                cin.ignore();
                getline(cin, name);
                do
                                {
                    cout << "Enter priority (0: Serious, 1: Non-serious, 2: General
Checkup): ";
                    cin >> priority;
                } while (priority < 0 || priority > 2);

                switch (priority)
                                {
                    case 0: pq.enqueue(name, SERIOUS); break;
                    case 1: pq.enqueue(name, NONSERIOUS); break;
                    case 2: pq.enqueue(name, CHECKUP); break;
                }
                break;

            case 2:
                pq.dequeue();
                break;

            case 3:
                pq.display();
                break;

            case 0:
                cout << "Exiting...\n";
                break;

            default:
                cout << "Invalid choice! Try again.\n";
```

```
169             }
170         } while (choice != 0);
171
172         return 0;
173     }
174
```

Output :




Select D:\SE Computer\LAB CODES\DSA\DSA10.exe
Hospital Priority Queue System
1. Add Patient
2. Serve Patient
3. Display Patients
0. Exit
Enter your choice: 3
Patient Queue (Higher priority first):
Bob - Non-serious
Alice - General Checkup
Hospital Priority Queue System
1. Add Patient
2. Serve Patient
3. Display Patients
0. Exit
Enter your choice: 2
Serving Patient: Bob (Priority: Non-serious)
Hospital Priority Queue System
1. Add Patient
2. Serve Patient
3. Display Patients
0. Exit
Enter your choice: 2
Serving Patient: Alice (Priority: General Checkup)
Hospital Priority Queue System
1. Add Patient
2. Serve Patient
3. Display Patients
0. Exit
Enter your choice: 3
No patients in queue.
Hospital Priority Queue System
1. Add Patient
2. Serve Patient
3. Display Patients
0. Exit
Enter your choice: 0
Exiting...
Process exited after 79.53 seconds with return value 0
Press any key to continue . . .