CHANDIGARH UNIVERSITY

Discover. Learn. Empower.

UNIVERSITY INSTITUTE OF COMPUTING

Project

Hospital Queue System

Submitted By:

Name: Paras Jain

UID: 24bca10454

Class & section: 24bca7(A)

Supervision By:

Name: Monika Choudhary

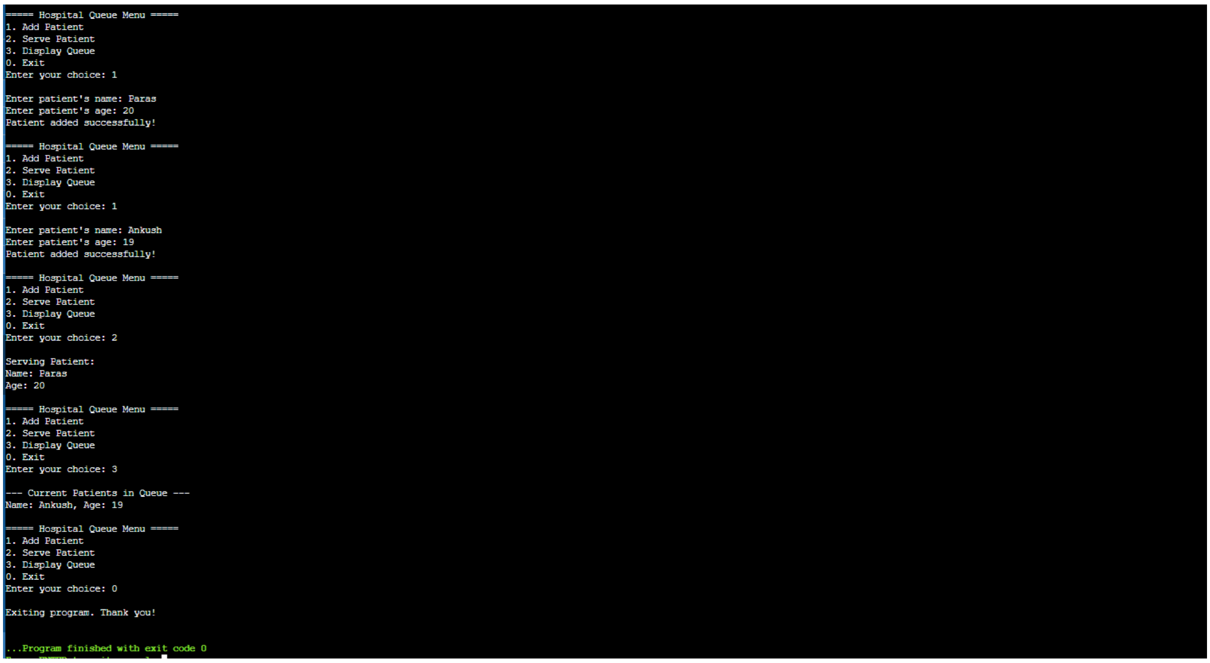
Designation: Assistant Professor

AIM :- To Develop Hospital Queue System Using Data Structure in C

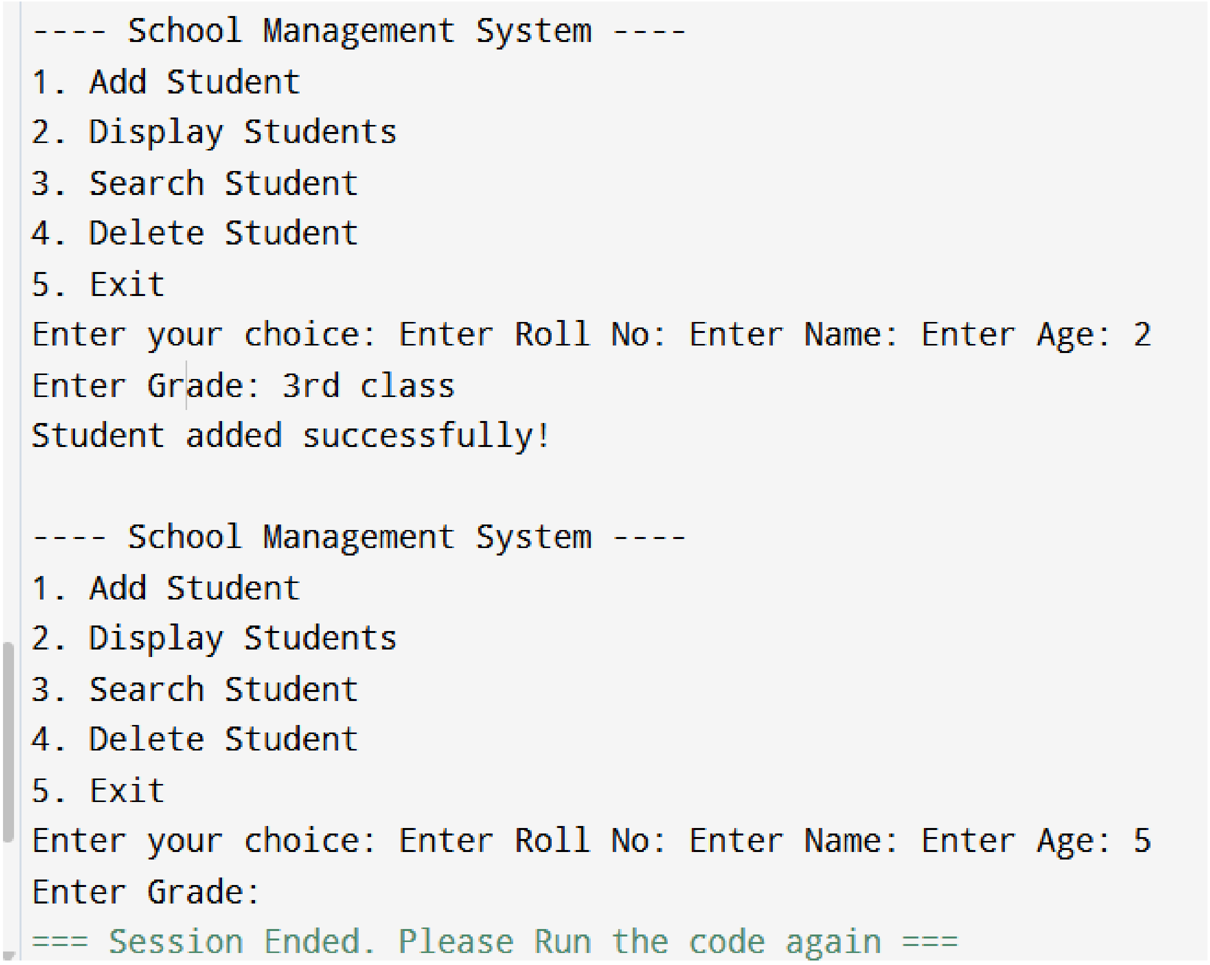
Procedure Code:-

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

#define SIZE 100

struct Patient {
    char name[50];
    int age;
};

struct Patient queue[SIZE];
int front = -1, rear = -1;

// Function to add a patient
void addPatient() {
    if (rear == SIZE - 1) {
        printf("\nQueue is full! Cannot add more patients.\n");
        return;
    }

    if (front == -1) {
        front = 0;
    }

    rear++;

    printf("\nEnter patient's name: ");
    fflush(stdin); // Use this to clear input buffer in Turbo C
    scanf("%s", queue[rear].name); // Avoids gets()

    printf("Enter patient's age: ");
    scanf("%d", &queue[rear].age);

    printf("Patient added successfully!\n");
}

// Function to serve the next patient
void servePatient() {
    if (front == -1 || front > rear) {
        printf("\nNo patients in the queue to serve.\n");
        return;
    }

    printf("\nServing Patient:\n");
    printf("Name: %s\n", queue[front].name);
    printf("Age: %d\n", queue[front].age);
    front++;
    if (front > rear) {
        front = rear = -1;
    }
}

// Function to display all patients in queue
void displayQueue() {
    if (front == -1 || front > rear) {
        printf("\nNo patients in the queue.\n");
        return;
    }

    printf("\n--- Current Patients in Queue ---\n");
    for (int i = front; i <= rear; i++) {
        printf("Name: %s, Age: %d\n", queue[i].name, queue[i].age);
    }
}

int main() {
    int choice;

    while (1) {
        printf("\n===== Hospital Queue Menu =====\n");
        printf("1. Add Patient\n");
        printf("2. Serve Patient\n");
        printf("3. Display Queue\n");
        printf("0. Exit\n");
        printf("Enter your choice: ");
        scanf("%d", &choice);

        switch (choice) {
            case 1:
                addPatient();
                break;
            case 2:
                servePatient();
                break;
            case 3:
                displayQueue();
                break;
            case 0:
                printf("\nExiting program. Thank you!\n");
                exit(0);
            default:
                printf("\nInvalid choice! Please try again.\n");
        }
    }

    return 0;
}
```

OUTPUT



```
===== Hospital Queue Menu =====
1. Add Patient
2. Serve Patient
3. Display Queue
0. Exit
Enter your choice: 1

Enter patient's name: Paras
Enter patient's age: 20
Patient added successfully!

===== Hospital Queue Menu =====
1. Add Patient
2. Serve Patient
3. Display Queue
0. Exit
Enter your choice: 1

Enter patient's name: Ankush
Enter patient's age: 19
Patient added successfully!

===== Hospital Queue Menu =====
1. Add Patient
2. Serve Patient
3. Display Queue
0. Exit
Enter your choice: 2

Serving Patient:
Name: Paras
Age: 20

===== Hospital Queue Menu =====
1. Add Patient
2. Serve Patient
3. Display Queue
0. Exit
Enter your choice: 3

--- Current Patients in Queue ---
Name: Ankush, Age: 19

===== Hospital Queue Menu =====
1. Add Patient
2. Serve Patient
3. Display Queue
0. Exit
Enter your choice: 0

Exiting program. Thank you!


...Program finished with exit code 0
```



```
---- School Management System ----
1. Add Student
2. Display Students
3. Search Student
4. Delete Student
5. Exit
Enter your choice: Enter Roll No: Enter Name: Enter Age: 2
Enter Grade: 3rd class
Student added successfully!

---- School Management System ----
1. Add Student
2. Display Students
3. Search Student
4. Delete Student
5. Exit
Enter your choice: Enter Roll No: Enter Name: Enter Age: 5
Enter Grade:
=== Session Ended. Please Run the code again ===
```