****

**Green University of Bangladesh**

**Department of Computer Science and Engineering (CSE)**

**Assignment NO – 04**

**Course Title: DS Section: D2**

**Submission Date : 12/2/2024**

**Course Teacher’s Name: Prof. Dr. Saiful Azad**

**Name: Rukonuzzaman Topu**

**ID : 232002280**

**Code:**

#include <stdio.h>

#include <string.h>

#include <stdbool.h>

int front = -1;

int rear = -1;

// Structure for student information

typedef struct {

char name[100];

char purpose[100];

} student;

// Structure for line database

typedef struct {

student stuData[100];

} dataBase;

// Function prototypes

void printMainMenu();

void enQueue(dataBase \*db);

void deQueue(dataBase \*db);

void printDb(dataBase \*db);

void countStudents();

// Check if the queue is empty

bool isEmpty() {

return front == -1;

}

// Check if the queue is full

bool isFull() {

return rear == 100 - 1;

}

int main() {

dataBase db;

for (;;) {

printMainMenu();

int choice;

printf("Enter your choice: ");

scanf("%d", &choice);

switch (choice) {

case 1:

enQueue(&db);

break;

case 2:

deQueue(&db);

break;

case 3:

printDb(&db);

break;

case 4:

countStudents();

break;

case 5:

return 0;

default:

printf("Invalid choice. Please try again.\n");

break;

}

}

return 0;

}

// Main menu function

void printMainMenu() {

printf("\n<< Main Menu >>\n");

printf("1. Add Student\n");

printf("2. Serve Student\n");

printf("3. View Current Student List\n");

printf("4. Total Students in Line\n");

printf("5. Exit\n");

return;

}

// Add a student to the queue

void enQueue(dataBase \*db) {

char name[100];

char purpose[100];

if (isFull()) {

printf("The line is full.\n");

return;

}

if (isEmpty()) {

front = 0;

rear = 0;

}

else {

rear++;

}

printf("Enter Name: ");

scanf(" %[^\n]", name);

printf("Enter Purpose: ");

scanf(" %[^\n]", purpose);

student newStudent;

strcpy(newStudent.name, name);

strcpy(newStudent.purpose, purpose);

db->stuData[rear] = newStudent;

printf("Added successfully.\n");

return;

}

// Serve (remove) a student from the queue

void deQueue(dataBase \*db) {

if (isEmpty()) {

printf("The line is empty.\n");

return;

}

printf("Served: %s (%s).\n", db->stuData[front].name, db->stuData[front].purpose);

if (front == rear) {

front = -1;

rear = -1;

}

else {

front++;

}

return;

}

// Display the list of students in the queue

void printDb(dataBase \*db) {

if (isEmpty()) {

printf("The line is Empty.\n");

return;

}

int j = 1;

printf("Students in Line:\n");

for (int i = front; i <= rear; i++) {

printf("%d. %s (%s)\n", j, db->stuData[i].name, db->stuData[i].purpose);

j++;

}

return;

}

// Count the number of students in the queue

void countStudents() {

if (isEmpty()) {

printf("The line is Empty.\n");

return;

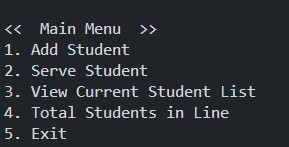
}

printf("Total Students in Line: %d\n", rear - front + 1);

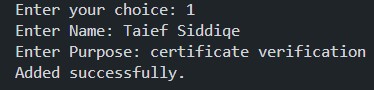
}

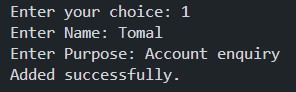
**Output:**

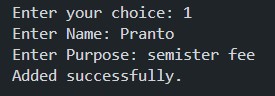
Step 1:

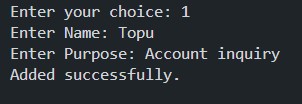


Step 2:

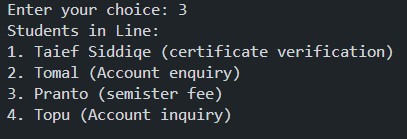




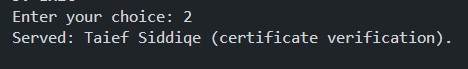




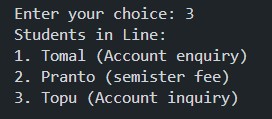
Step 3:



Step 4:

****

Step 5:



Step 6:



Step 7:

