Republic of the Philippines
**PAMANTASAN NG LUNGSOD NG MAYNILA**
Intramuros, Manila, Philippines

# College of Information Systems and Technology Management

## 2nd Semester A.Y. 2023-2024

## ICC 0104-1 DSA

**GROUP 4**

**Submitted by: Leader**
Garsota, Janine B.

**Members**
Callangan, Moira C.
Capiral, Luis Gabriel A.
Corales, Charliz Dana M.
Goyena, Shawn Kieffer E.
King, Mariano Luiz B.
Mabutas, Carla R.
Pajarito, Rose Krishna Cassandra
Verdida, Maverick Isaiah A.

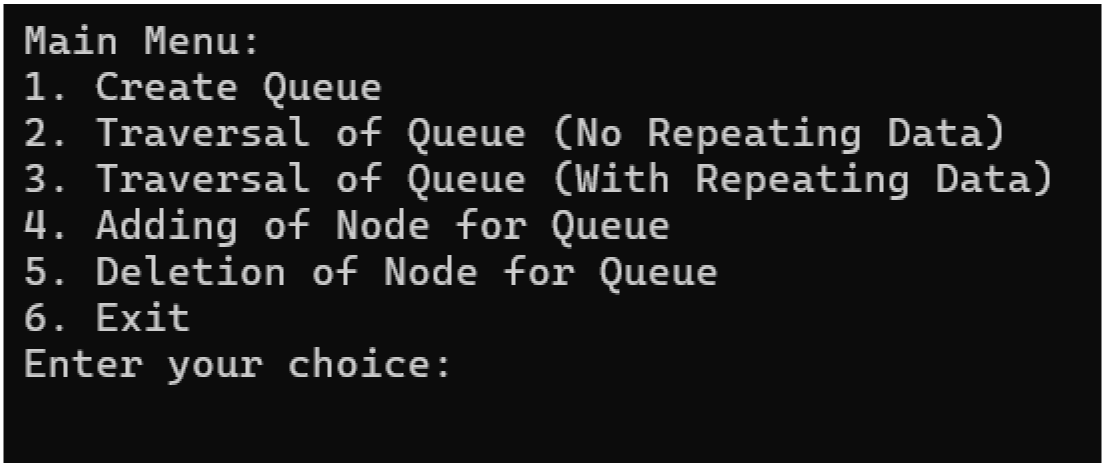
**Submitted to:**
PROF. RICHARD C. REGALA

## Converted Queues from C to C++

```
#include <iostream>
#include <string>
#include <queue>
#include <windows.h>

struct STUDREC {
    std::string No;
    std::string Name;
    std::string Cyr;
    float GWA;
};

std::queue<STUDREC> studentQueue;

void createQueue();
void traversalNoRepeating();
void traversalWithRepeating();
void addNode();
void deleteNode();
void gotoxy(short x, short y);

int main() {
    int choice;

    do {
        system("cls");
        std::cout << "Main Menu:\n";
        std::cout << "1. Create Queue\n";
        std::cout << "2. Traversal of Queue (No Repeating Data)\n";
        std::cout << "3. Traversal of Queue (With Repeating Data)\n";
        std::cout << "4. Adding of Node for Queue\n";
        std::cout << "5. Deletion of Node for Queue\n";
        std::cout << "6. Exit\n";
        std::cout << "Enter your choice: ";
        std::cin >> choice;

        switch (choice) {
            case 1:
                createQueue();
                break;
            case 2:
                traversalNoRepeating();
                break;
            case 3:
                traversalWithRepeating();
                break;
            case 4:
                addNode();
                break;
            case 5:
                deleteNode();
                break;
            case 6:
                exit(0);
            default:
                std::cout << "Invalid choice! Please try again.\n";
        }

        std::cout << "\nPress any key to continue...";
        std::cin.ignore();
        std::cin.get();
    } while (1);

    return 0;
}

void createQueue() {
    if (!studentQueue.empty()) {
        std::cout << "You already have queue/s.\n";
        return;
    }

    char choice;
    do {
        STUDREC newNode;

        std::cout << "\nEnter student No: ";
        std::cin >> newNode.No;
        std::cout << "Enter student Name: ";
        std::cin.ignore();
        std::getline(std::cin, newNode.Name);
        std::cout << "Enter student Course and Year: ";
        std::getline(std::cin, newNode.Cyr);
        std::cout << "Enter student GWA: ";
        std::cin >> newNode.GWA;

        studentQueue.push(newNode);
        std::cout << "\nDo you want to add more records? [Y/N]: ";
        std::cin >> choice;
    } while (choice == 'Y' || choice == 'y');
}

void traversalNoRepeating() {
    if (studentQueue.empty()) {
        std::cout << "The queue is empty!\n";
        return;
    }

    system("cls");

    std::queue<STUDREC> tempQueue = studentQueue;
    int row = 0;
    bool repeated = false;
    std::string Flag;

    std::cout << "Student Records:\n";
    gotoxy(0, 2 + row); std::cout << "Student Number";
    gotoxy(25, 2 + row); std::cout << "Student Name";
    gotoxy(50, 2 + row); std::cout << "Student Cyr";
    gotoxy(75, 2 + row); std::cout << "Student GWA";

    if (!tempQueue.empty()) {
        STUDREC TravNode = tempQueue.front();
        tempQueue.pop();

        gotoxy(0, 3 + row); std::cout << TravNode.No;
        gotoxy(25, 3 + row); std::cout << TravNode.Name;
        gotoxy(50, 3 + row); std::cout << TravNode.Cyr;
        gotoxy(75, 3 + row); std::cout << TravNode.GWA;
        row++;

        Flag = TravNode.No;

        while (!tempQueue.empty() && !repeated) {
            TravNode = tempQueue.front();
            tempQueue.pop();

            if (Flag != TravNode.No) {
                gotoxy(0, 3 + row); std::cout << TravNode.No;
                gotoxy(25, 3 + row); std::cout << TravNode.Name;
                gotoxy(50, 3 + row); std::cout << TravNode.Cyr;
                gotoxy(75, 3 + row); std::cout << TravNode.GWA;
                row++;
            } else {
                repeated = true;
            }
        }
    }

    if (repeated) {
        std::cout << "\nTraversal stopped.\n";
    }
}

void traversalWithRepeating() {
    if (studentQueue.empty()) {
        std::cout << "The queue is empty!\n";
        return;
    }

    system("cls");

    std::queue<STUDREC> tempQueue;
    int row = 0;

    std::cout << "Student Records:\n";
    gotoxy(0, 2 + row); std::cout << "Student Number";
    gotoxy(25, 2 + row); std::cout << "Student Name";
    gotoxy(50, 2 + row); std::cout << "Student Cyr";
    gotoxy(75, 2 + row); std::cout << "Student GWA";

    while (!studentQueue.empty()) {
        STUDREC TravNode = studentQueue.front();
        studentQueue.pop();

        gotoxy(0, 3 + row); std::cout << TravNode.No;
        gotoxy(25, 3 + row); std::cout << TravNode.Name;
        gotoxy(50, 3 + row); std::cout << TravNode.Cyr;
        gotoxy(75, 3 + row); std::cout << TravNode.GWA;
        row++;

        tempQueue.push(TravNode);
    }
    studentQueue = tempQueue;
}

void addNode() {
    STUDREC newNode;

    std::cout << "\nEnter student No: ";
    std::cin >> newNode.No;
    std::cout << "Enter student Name: ";
    std::cin.ignore();
    std::getline(std::cin, newNode.Name);
    std::cout << "Enter student Course and Year: ";
    std::getline(std::cin, newNode.Cyr);
    std::cout << "Enter student GWA: ";
    std::cin >> newNode.GWA;

    studentQueue.push(newNode);
}

void deleteNode() {
    if (studentQueue.empty()) {
        std::cout << "The queue is empty!\n";
        return;
    }

    studentQueue.pop();
    std::cout << "Node deleted successfully.\n";
}

void gotoxy(short x, short y) {
    COORD pos = {x, y};
    SetConsoleCursorPosition(GetStdHandle(STD_OUTPUT_HANDLE), pos);
}
```

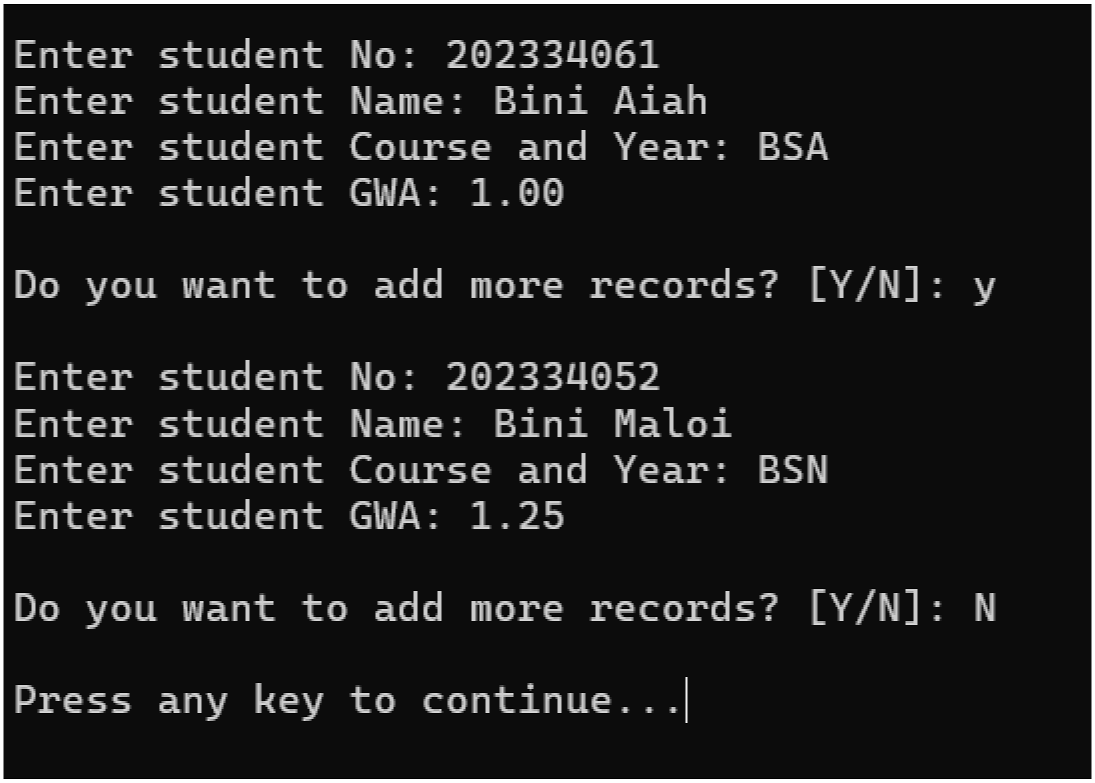
**Sample Outputs**

1. Main Menu



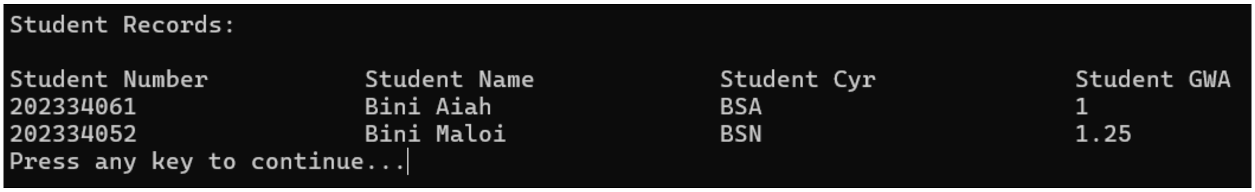
```
Main Menu:
1. Create Queue
2. Traversal of Queue (No Repeating Data)
3. Traversal of Queue (With Repeating Data)
4. Adding of Node for Queue
5. Deletion of Node for Queue
6. Exit
Enter your choice:
```

2. Create Queue



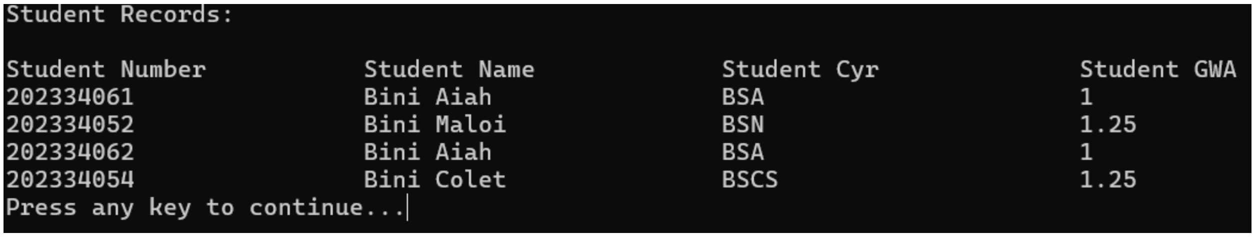
```
Enter student No: 202334061
Enter student Name: Bini Aiah
Enter student Course and Year: BSA
Enter student GWA: 1.00

Do you want to add more records? [Y/N]: y

Enter student No: 202334052
Enter student Name: Bini Maloi
Enter student Course and Year: BSN
Enter student GWA: 1.25

Do you want to add more records? [Y/N]: N

Press any key to continue...
```

3. Traversal of Queue (No Repeating Data)



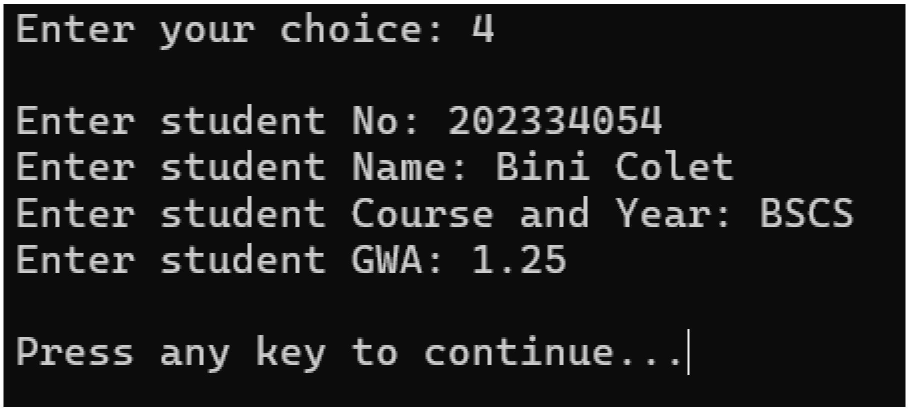
```
Student Records:

Student Number           Student Name             Student Cyr              Student GWA
202334061                Bini Aiah                BSA                      1
202334052                Bini Maloi               BSN                      1.25
Press any key to continue...
```

4. Traversal of Queue (With Repeating Data)



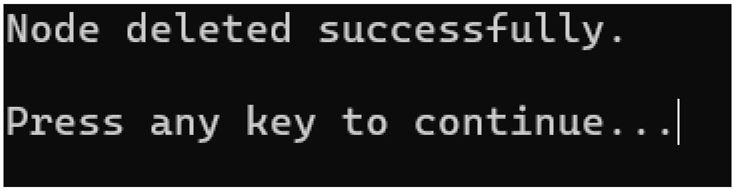
```
Student Records:

Student Number           Student Name             Student Cyr              Student GWA
202334061                Bini Aiah                BSA                      1
202334052                Bini Maloi               BSN                      1.25
202334062                Bini Aiah                BSA                      1
202334054                Bini Colet               BSCS                     1.25
Press any key to continue...
```

5. Adding of node for Queue

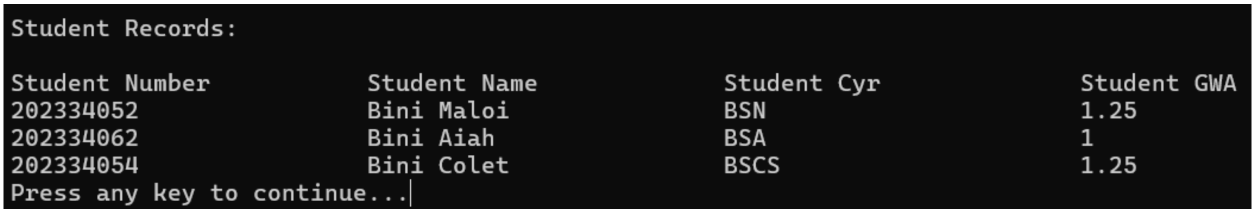


```
Enter your choice: 4

Enter student No: 202334054
Enter student Name: Bini Colet
Enter student Course and Year: BSCS
Enter student GWA: 1.25

Press any key to continue...
```

6. Deletion of node for Queue

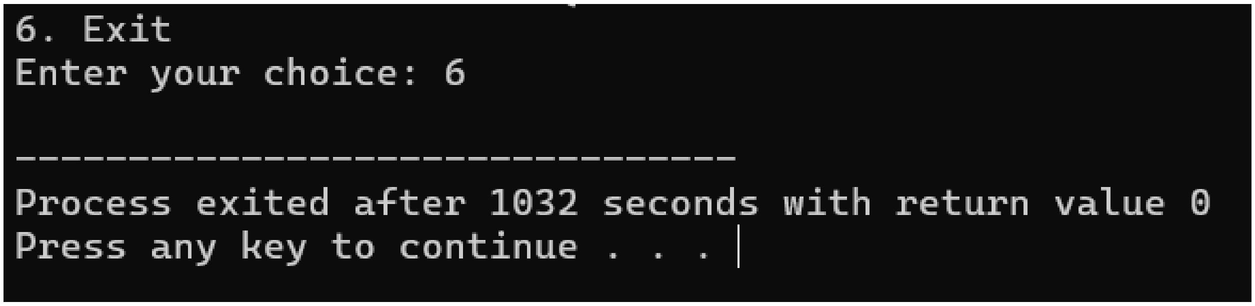


```
Node deleted successfully.

Press any key to continue...
```



```
Student Records:

Student Number           Student Name             Student Cyr              Student GWA
202334052                Bini Maloi               BSN                      1.25
202334062                Bini Aiah                BSA                      1
202334054                Bini Colet               BSCS                     1.25
Press any key to continue...
```

7. Exit



```
6. Exit
Enter your choice: 6


--------------------------------
Process exited after 1032 seconds with return value 0
Press any key to continue . . .
```