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 Billones

Members
Callangan, Moira
Capiral, Luis
Corales, Charliz Dana
Goyena, Shawn Kieffer
King, Mariano Luiz
Mabutas, Carla
Pajarito, Rose Krishna
Verdida, Maverick

Submitted to:

PROF. RICHARD C. REGALA

Republic of the Philippines

PAMANTASAN NG LUNGSOD NG MAYNILA

Intramuros, Manila, Philippines

SOURCE CODE FOR QUEUES

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

typedef struct STUDREC {
    char No[13];
    char Name[25];
    char Cyr[10];
    float GWA;
    struct STUDREC *next;
} STUDREC;

STUDREC *FRONT = NULL;
STUDREC *REAR = NULL;

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

int main() {
    int choice;
    char ch;

    do {
        system("cls");
        printf("Main Menu:\n");
        printf("1. Create Queue\n");
        printf("2. Traversal of Queue (No Repeating Data)\n");
        printf("3. Traversal of Queue (With Repeating Data)\n");
        printf("4. Adding of Node for Queue\n");
        printf("5. Deletion of Node for Queue\n");
        printf("6. Exit\n");
        printf("Enter your choice: ");
        scanf("%d", &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:
                printf("Invalid choice! Please try again.\n");
        }

        printf("\nPress any key to continue...");
        getchar();
        getchar();
    } while (1);

    return 0;
}

void createQueue() {
    if (FRONT != NULL) {
        printf("You already have queue/s.\n");
        return;
    }

    char choice;
    do {
        STUDREC *newNode = (STUDREC*)malloc(sizeof(STUDREC));
        if (newNode == NULL) {
            printf("Memory allocation failed!\n");
            STUDREC *temp = FRONT;
```

```
            while (temp != NULL) {
                STUDREC *next = temp->next;
                free(temp);
                temp = next;
            }
            exit(1);
        }

        printf("\nEnter student No: ");
        scanf("%s", newNode->No);
        printf("Enter student Name: ");
        scanf(" %[^\n]", newNode->Name);
        printf("Enter student Course and Year: ");
        scanf(" %[^\n]", newNode->Cyr);
        printf("Enter student GWA: ");
        scanf("%f", &newNode->GWA);

        newNode->next = NULL;

        if (FRONT == NULL) {
            FRONT = REAR = newNode;
        } else {
            REAR->next = newNode;
            REAR = newNode;
        }

        printf("\nDo you want to add more records? [Y/N]: ");
        scanf(" %c", &choice);
    } while (choice == 'Y' || choice == 'y');
}

void traversalNoRepeating() {
    if (FRONT == NULL) {
        printf("The queue is empty!\n");
        return;
    }

    system("cls");

    STUDREC *TravNode = FRONT;
    int row = 0;
    int repeated = 0;
```

```
    printf("Student Records:");
    gotoxy(0, 2 + row); printf("Student Number");
    gotoxy(25, 2 + row); printf("Student Name");
    gotoxy(50, 2 + row); printf("Student Cyr");
    gotoxy(75, 2 + row); printf("Student GWA");


    gotoxy(0, 3 + row); printf("%s", TravNode->No);
    gotoxy(25, 3 + row); printf("%s", TravNode->Name);
    gotoxy(50, 3 + row); printf("%s", TravNode->Cyr);
    gotoxy(75, 3 + row); printf("%.2f", TravNode->GWA);
    row++;

    char Flag[13];
    strcpy(Flag, TravNode->No);

    TravNode = TravNode->next;

    while (TravNode != NULL && !repeated) {

        if (strcmp(Flag, TravNode->No) != 0) {
            gotoxy(0, 3 + row); printf("%s", TravNode->No);
            gotoxy(25, 3 + row); printf("%s", TravNode->Name);
            gotoxy(50, 3 + row); printf("%s", TravNode->Cyr);
            gotoxy(75, 3 + row); printf("%.2f", TravNode->GWA);
            row++;
        } else {
            repeated = 1;
        }

        TravNode = TravNode->next;
    }

    if (repeated) {
        printf("\nTraversal stopped.\n");
    }
}

void traversalWithRepeating() {
    if (FRONT == NULL) {
        printf("The queue is empty!\n");
        return;
```

```
}

system("cls");

STUDREC *TravNode = FRONT;
STUDREC *TempQueue = NULL;
STUDREC *TempQueueTail = NULL;

int row = 0;

printf("Student Records:");
gotoxy(0,2+row);  printf("Student Number");
gotoxy(25,2+row);         printf("Student Name");
gotoxy(50,2+row);         printf("Student Cyr");
gotoxy(75,2+row);         printf("Student GWA");

while (TravNode != NULL) {
   gotoxy(0,3+row); printf("%s", TravNode->No);
   gotoxy(25,3+row); printf("%s", TravNode->Name);
   gotoxy(50,3+row); printf("%s", TravNode->Cyr);
   gotoxy(75,3+row); printf("%.2f", TravNode->GWA);
   row++;


   STUDREC *newNode = (STUDREC*)malloc(sizeof(STUDREC));
   if (newNode == NULL) {
      printf("Memory allocation failed!\n");
      exit(1);
   }
   strcpy(newNode->No, TravNode->No);
   strcpy(newNode->Name, TravNode->Name);
   strcpy(newNode->Cyr, TravNode->Cyr);
   newNode->GWA = TravNode->GWA;
   newNode->next = NULL;

   if (TempQueue == NULL) {
      TempQueue = newNode;
      TempQueueTail = newNode;
   } else {
      TempQueueTail->next = newNode;
      TempQueueTail = newNode;
   }
```

```
        TravNode = TravNode->next;
    }


    while (FRONT != NULL) {
        STUDREC *temp = FRONT;
        FRONT = FRONT->next;
        free(temp);
    }
    REAR = NULL;

    FRONT = TempQueue;
    REAR = TempQueueTail;
}


void addNode() {
    STUDREC *newNode = (STUDREC*)malloc(sizeof(STUDREC));
    if (newNode == NULL) {
        printf("Memory allocation failed!\n");

        STUDREC *temp = FRONT;
        while (temp != NULL) {
            STUDREC *next = temp->next;
            free(temp);
            temp = next;
        }
        exit(1);
    }

    printf("\nEnter student No: ");
    scanf("%s", newNode->No);
    printf("Enter student Name: ");
    scanf(" %[^\n]", newNode->Name);
    printf("Enter student Course and Year: ");
    scanf(" %[^\n]", newNode->Cyr);
    printf("Enter student GWA: ");
    scanf("%f", &newNode->GWA);

    newNode->next = NULL;
```

```
    if (FRONT == NULL) {
        FRONT = REAR = newNode;
    } else {
        REAR->next = newNode;
        REAR = newNode;
    }
}

void deleteNode() {
    if (FRONT == NULL) {
        printf("The queue is empty!\n");
        return;
    }

    STUDREC *temp = FRONT;
    FRONT = FRONT->next;
    free(temp);
    printf("Node deleted successfully.\n");
}

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

Republic of the Philippines

PAMANTASAN NG LUNGSOD NG MAYNILA

Intramuros, Manila, Philippines

SAMPLE OUTPUT FOR QUEUES

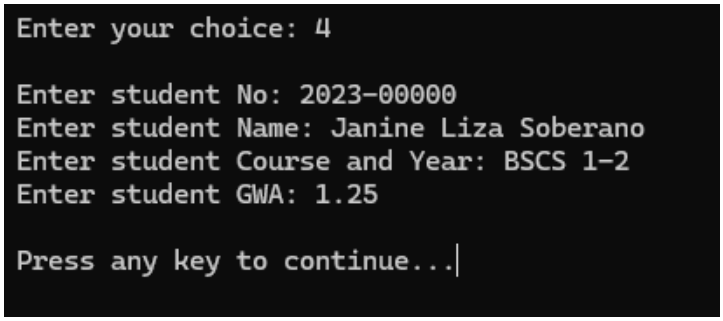
1. Creation of Queue

```
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: 1

Enter student No: 2023-00000
Enter student Name: Janine Cute
Enter student Course and Year: BSCS 1-1
Enter student GWA: 1.00

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

Enter student No: 2023-11111
Enter student Name: Ryan Gosling
Enter student Course and Year: BSCS 1-1
Enter student GWA: 1.00

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

Press any key to continue...
```

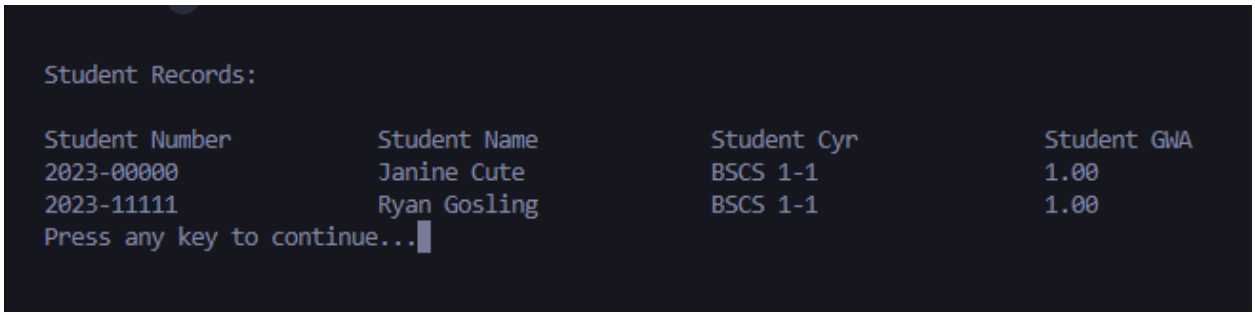
If you already have queues:

```
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: 1
You already have queue/s.

Press any key to continue...
```

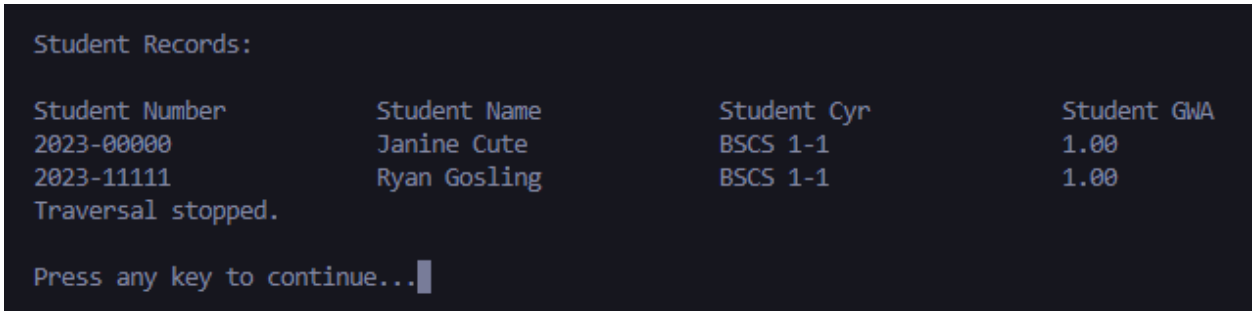
2. **Adding Node for Queue**



3. **Traversal of Queue (no repeating data)**
(without the application of step 2)



If there is already an existing student number (with the application of step 2)



4. **Traversal of Queue (with repeating data)**

(without the application of step 2)

```
Student Records:

Student Number          Student Name            Student Cyr             Student GWA
2023-00000              Janine Cute             BSCS 1-1                1.00
2023-11111              Ryan Gosling            BSCS 1-1                1.00
Press any key to continue...
```

If there is already an existing student number (with the application of step 2)

```
Student Records:

Student Number          Student Name            Student Cyr             Student GWA
2023-00000              Janine Cute             BSCS 1-1                1.00
2023-11111              Ryan Gosling            BSCS 1-1                1.00
2023-00000              Janine Liza Soberano    BSCS 1-2                1.25
Press any key to continue...
```

5. **Deletion of Node for Queue**

If there is no existing queue:

```
Enter your choice: 5
The queue is empty!

Press any key to continue...
```

If there is an existing queue for node in Traversal of Queue (No Repeating Data)

```
Student Records:

Student Number          Student Name            Student Cyr             Student GWA
2023-11111              Ryan Gosling            BSCS 1-1                1.00
2023-00000              Janine Liza Soberano    BSCS 1-2                1.25
Press any key to continue...
```

If there is an existing queue for node in Traversal of Queue (With Repeating Data)

```
Student Records:

Student Number          Student Name            Student Cyr             Student GWA
2023-11111              Ryan Gosling            BSCS 1-1                1.00
2023-00000              Janine Liza Soberano    BSCS 1-2                1.25
Press any key to continue...
```

6. **Exit**

```
Enter your choice: 6

--------------------------------
Process exited after 770 seconds with return value 0
Press any key to continue . . .
```