PAMANTASAN NG LUNGSOD NG MAYNILA

College of Information Systems and Technology Management (CISTM)

ICC 0104-1 – Data Structures and Algorithms

A.Y. 2023- 2024

Group 4: Searching

Submitted by:

Abundo, Jonalene Ryza B.

Dela Peña, Daniella Mae N.

Diamzon, Momer Ailes M.

Lau, Trisha Mae R.

Mahusay, Lindsay G.

Matanga, Sophia Vien V.

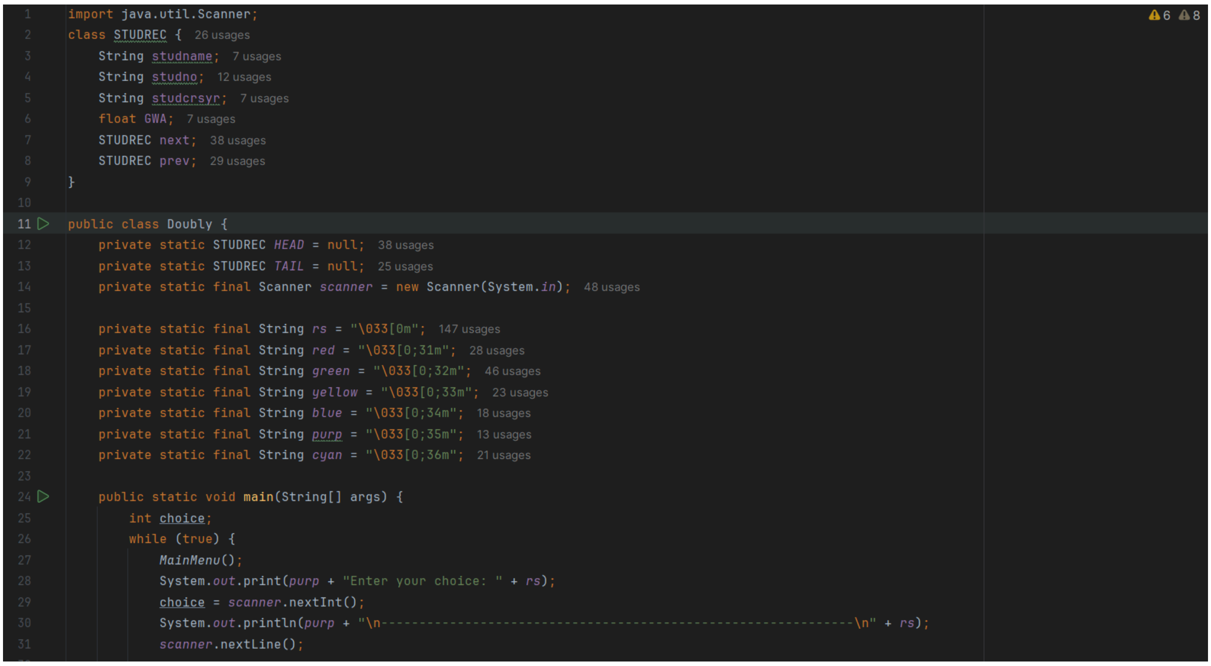
Rivera, Ramyll C.

Sibayan, Joan F.

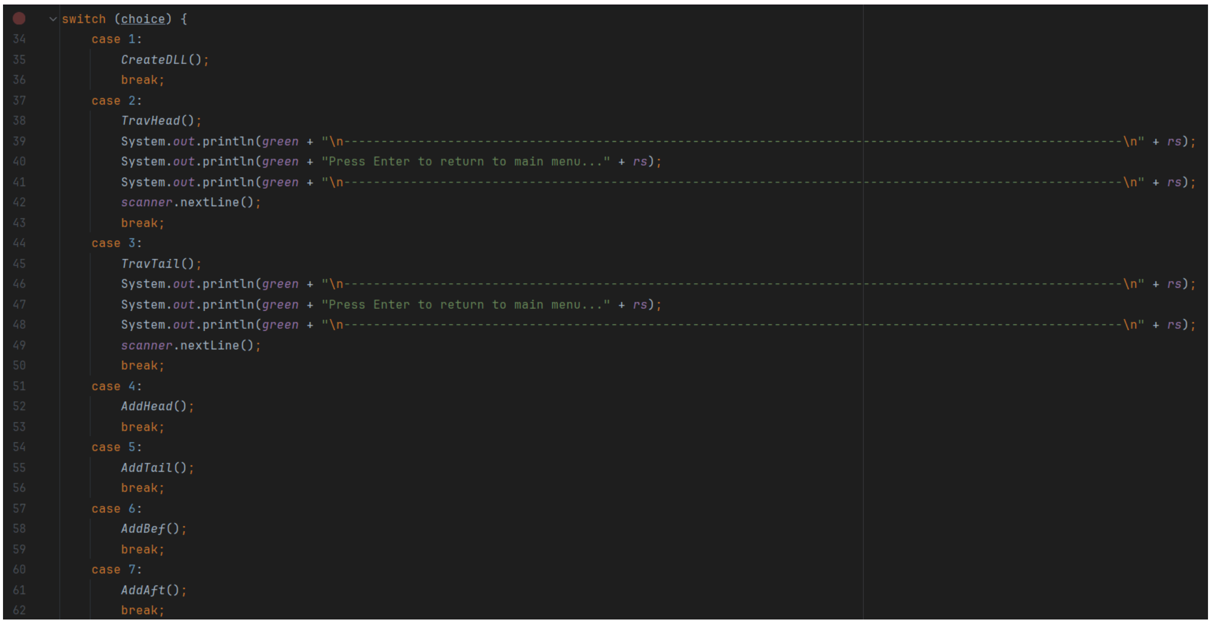
BSCS 1-1

I. Main Menu

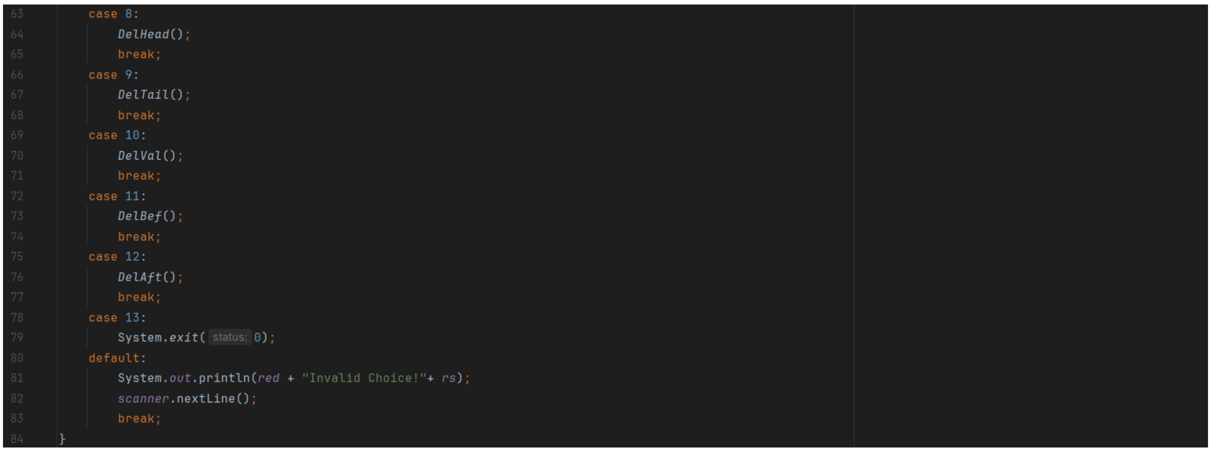
Source Code in Java



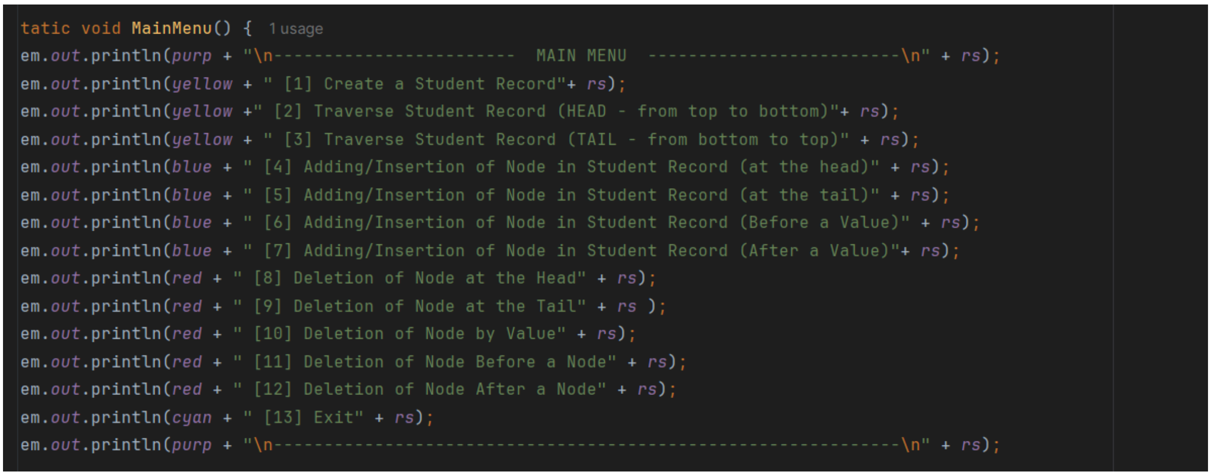
```java
import java.util.Scanner;
class STUDREC {  26 usages
    String studname;  7 usages
    String studno;  12 usages
    String studcrsyr;  7 usages
    float GWA;  7 usages
    STUDREC next;  38 usages
    STUDREC prev;  29 usages
}

public class Doubly {
    private static STUDREC HEAD = null;  38 usages
    private static STUDREC TAIL = null;  25 usages
    private static final Scanner scanner = new Scanner(System.in);  48 usages

    private static final String rs = "\033[0m";  147 usages
    private static final String red = "\033[0;31m";  28 usages
    private static final String green = "\033[0;32m";  46 usages
    private static final String yellow = "\033[0;33m";  23 usages
    private static final String blue = "\033[0;34m";  18 usages
    private static final String purp = "\033[0;35m";  13 usages
    private static final String cyan = "\033[0;36m";  21 usages

    public static void main(String[] args) {
        int choice;
        while (true) {
            MainMenu();
            System.out.print(purp + "Enter your choice: " + rs);
            choice = scanner.nextInt();
            System.out.println(purp + "\n--------------------------------------------------------------\n" + rs);
            scanner.nextLine();
```



```java
switch (choice) {
    case 1:
        CreateDLL();
        break;
    case 2:
        TravHead();
        System.out.println(green + "\n---------------------------------------------------------------------------------------------------------\n" + rs);
        System.out.println(green + "Press Enter to return to main menu..." + rs);
        System.out.println(green + "\n---------------------------------------------------------------------------------------------------------\n" + rs);
        scanner.nextLine();
        break;
    case 3:
        TravTail();
        System.out.println(green + "\n---------------------------------------------------------------------------------------------------------\n" + rs);
        System.out.println(green + "Press Enter to return to main menu..." + rs);
        System.out.println(green + "\n---------------------------------------------------------------------------------------------------------\n" + rs);
        scanner.nextLine();
        break;
    case 4:
        AddHead();
        break;
    case 5:
        AddTail();
        break;
    case 6:
        AddBef();
        break;
    case 7:
        AddAft();
        break;
```

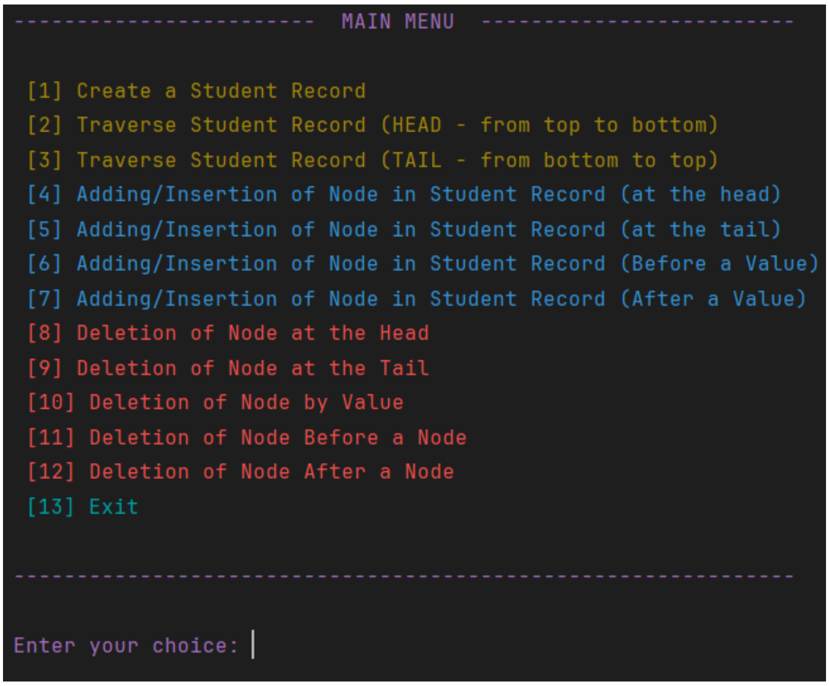


```
    case 8:
        DelHead();
        break;
    case 9:
        DelTail();
        break;
    case 10:
        DelVal();
        break;
    case 11:
        DelBef();
        break;
    case 12:
        DelAft();
        break;
    case 13:
        System.exit( status: 0);
    default:
        System.out.println(red + "Invalid Choice!"+ rs);
        scanner.nextLine();
        break;
}
```



```
tatic void MainMenu() {  1 usage
em.out.println(purp + "\n------------------------  MAIN MENU  -------------------------\n" + rs);
em.out.println(yellow + " [1] Create a Student Record"+ rs);
em.out.println(yellow +" [2] Traverse Student Record (HEAD - from top to bottom)"+ rs);
em.out.println(yellow + " [3] Traverse Student Record (TAIL - from bottom to top)" + rs);
em.out.println(blue + " [4] Adding/Insertion of Node in Student Record (at the head)" + rs);
em.out.println(blue + " [5] Adding/Insertion of Node in Student Record (at the tail)" + rs);
em.out.println(blue + " [6] Adding/Insertion of Node in Student Record (Before a Value)" + rs);
em.out.println(blue + " [7] Adding/Insertion of Node in Student Record (After a Value)"+ rs);
em.out.println(red + " [8] Deletion of Node at the Head" + rs);
em.out.println(red + " [9] Deletion of Node at the Tail" + rs );
em.out.println(red + " [10] Deletion of Node by Value" + rs);
em.out.println(red + " [11] Deletion of Node Before a Node" + rs);
em.out.println(red + " [12] Deletion of Node After a Node" + rs);
em.out.println(cyan + " [13] Exit" + rs);
em.out.println(purp + "\n--------------------------------------------------------------\n" + rs);
```

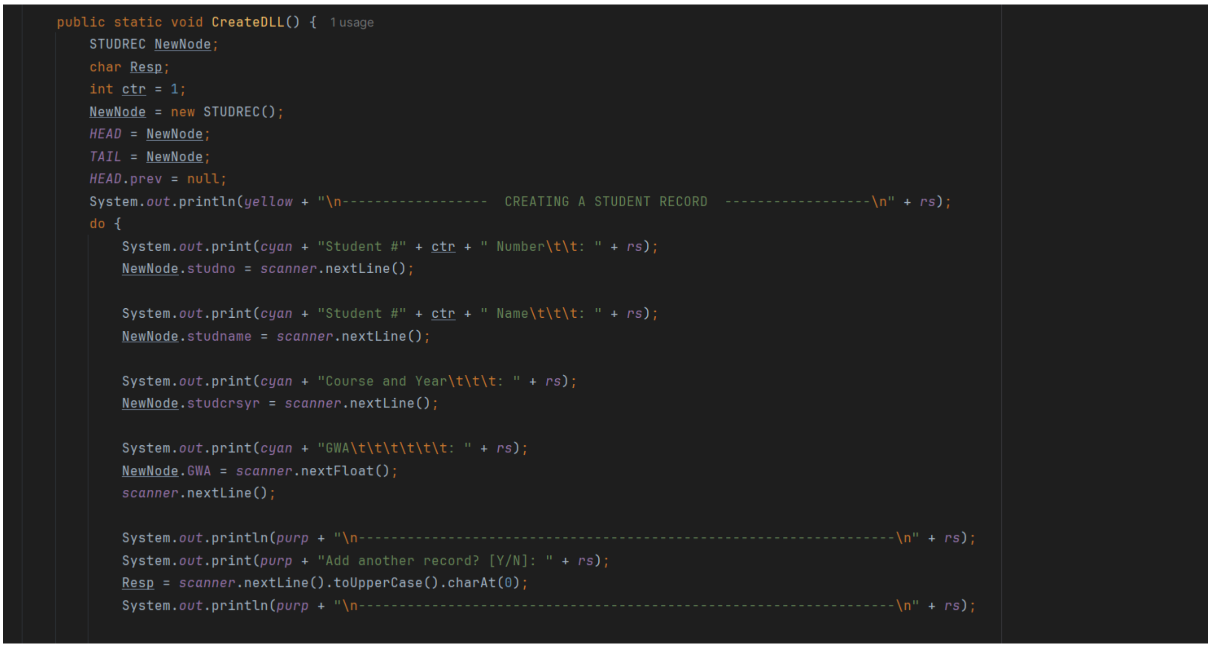
Output



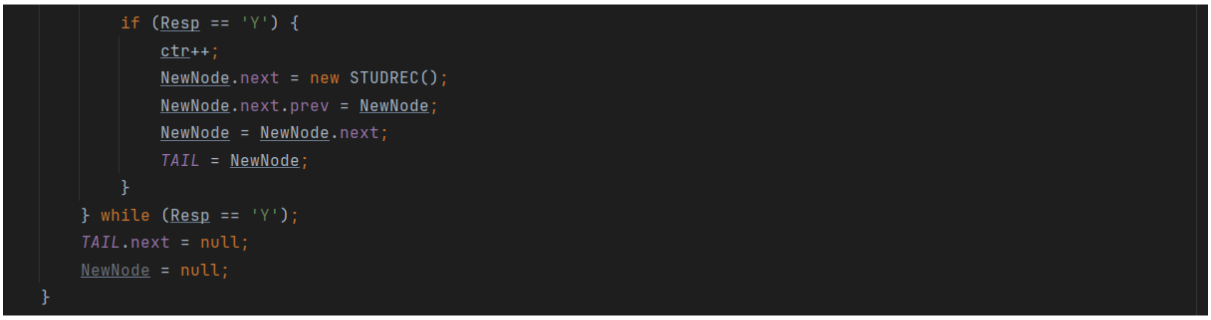
```
------------------------  MAIN MENU  -------------------------

 [1] Create a Student Record
 [2] Traverse Student Record (HEAD - from top to bottom)
 [3] Traverse Student Record (TAIL - from bottom to top)
 [4] Adding/Insertion of Node in Student Record (at the head)
 [5] Adding/Insertion of Node in Student Record (at the tail)
 [6] Adding/Insertion of Node in Student Record (Before a Value)
 [7] Adding/Insertion of Node in Student Record (After a Value)
 [8] Deletion of Node at the Head
 [9] Deletion of Node at the Tail
 [10] Deletion of Node by Value
 [11] Deletion of Node Before a Node
 [12] Deletion of Node After a Node
 [13] Exit


--------------------------------------------------------------


Enter your choice: 
```

II. Creation of Double/Doubly Linked List

Source Code in Java

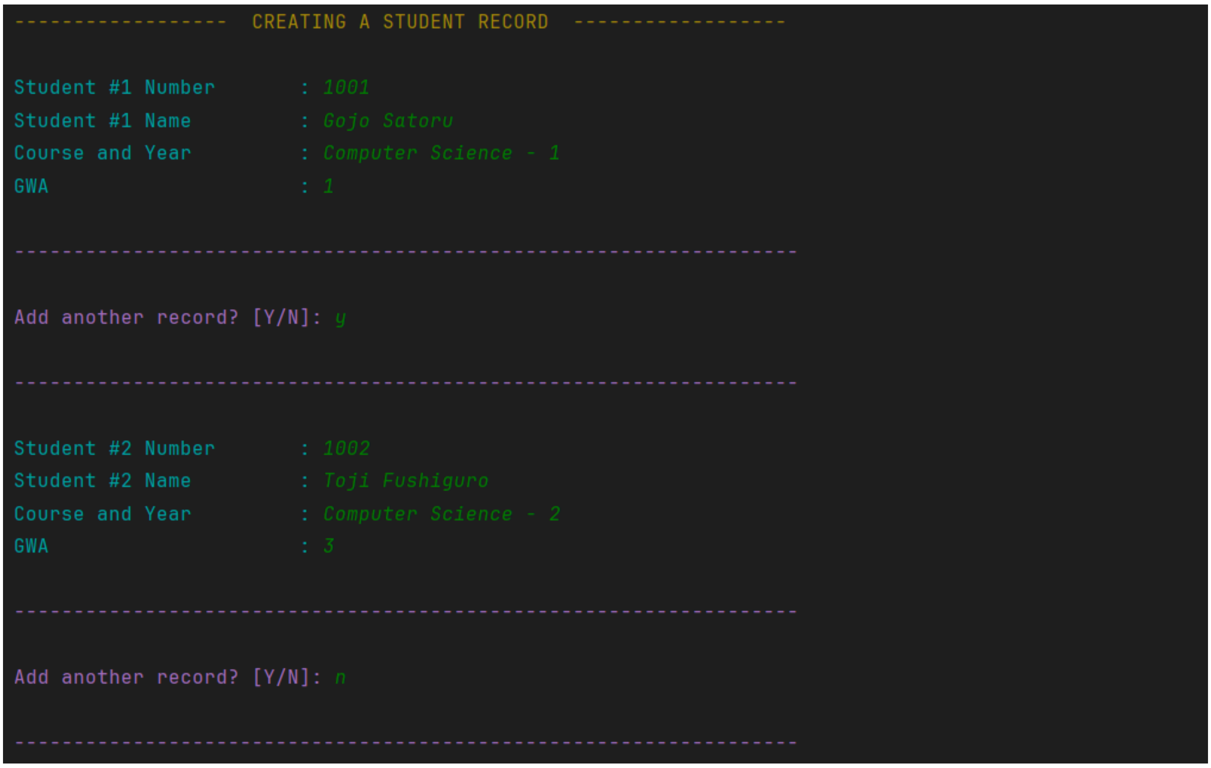


```java
public static void CreateDLL() {  1 usage
    STUDREC NewNode;
    char Resp;
    int ctr = 1;
    NewNode = new STUDREC();
    HEAD = NewNode;
    TAIL = NewNode;
    HEAD.prev = null;
    System.out.println(yellow + "\n------------------  CREATING A STUDENT RECORD  ------------------\n" + rs);
    do {
        System.out.print(cyan + "Student #" + ctr + " Number\t\t: " + rs);
        NewNode.studno = scanner.nextLine();

        System.out.print(cyan + "Student #" + ctr + " Name\t\t\t: " + rs);
        NewNode.studname = scanner.nextLine();

        System.out.print(cyan + "Course and Year\t\t\t: " + rs);
        NewNode.studcrsyr = scanner.nextLine();

        System.out.print(cyan + "GWA\t\t\t\t\t\t: " + rs);
        NewNode.GWA = scanner.nextFloat();
        scanner.nextLine();

        System.out.println(purp + "\n------------------------------------------------------------------\n" + rs);
        System.out.print(purp + "Add another record? [Y/N]: " + rs);
        Resp = scanner.nextLine().toUpperCase().charAt(0);
        System.out.println(purp + "\n------------------------------------------------------------------\n" + rs);
```



```java
        if (Resp == 'Y') {
            ctr++;
            NewNode.next = new STUDREC();
            NewNode.next.prev = NewNode;
            NewNode = NewNode.next;
            TAIL = NewNode;
        }
    } while (Resp == 'Y');
    TAIL.next = null;
    NewNode = null;
}
```

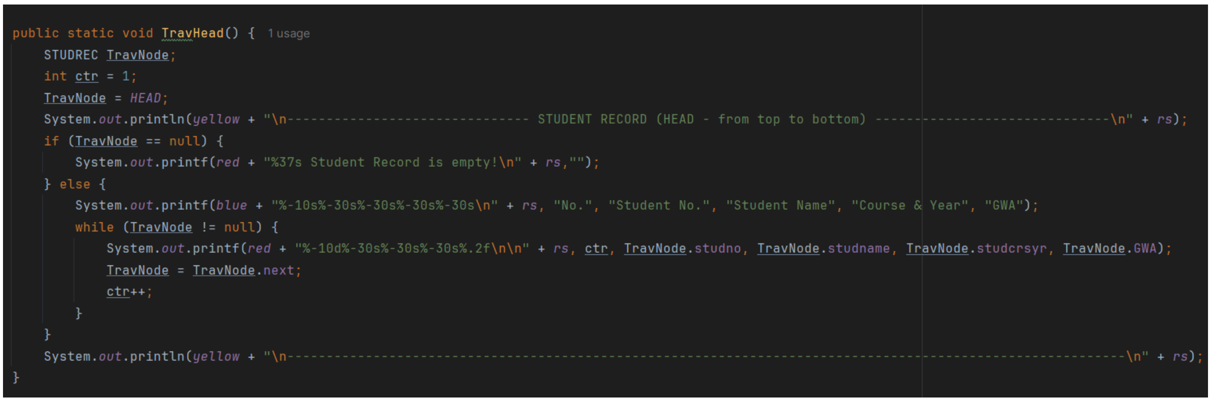
Output



```
------------------  CREATING A STUDENT RECORD  ------------------

Student #1 Number       : 1001
Student #1 Name         : Gojo Satoru
Course and Year         : Computer Science - 1
GWA                     : 1

------------------------------------------------------------------

Add another record? [Y/N]: y

------------------------------------------------------------------

Student #2 Number       : 1002
Student #2 Name         : Toji Fushiguro
Course and Year         : Computer Science - 2
GWA                     : 3

------------------------------------------------------------------

Add another record? [Y/N]: n

------------------------------------------------------------------
```

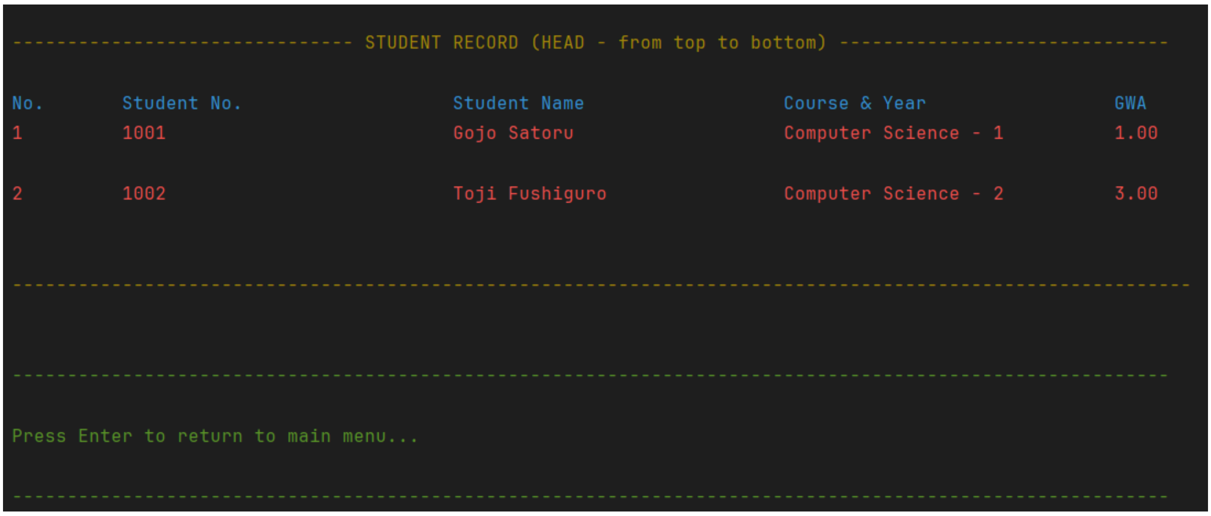
III. Traversal of Double/Doubly Linked List (Head - Top to Bottom)

Source Code in Java



```java
public static void TravHead() {  1 usage
    STUDREC TravNode;
    int ctr = 1;
    TravNode = HEAD;
    System.out.println(yellow + "\n------------------------------ STUDENT RECORD (HEAD - from top to bottom) ------------------------------\n" + rs);
    if (TravNode == null) {
        System.out.printf(red + "%37s Student Record is empty!\n" + rs,"");
    } else {
        System.out.printf(blue + "%-10s%-30s%-30s%-30s%-30s\n" + rs, "No.", "Student No.", "Student Name", "Course & Year", "GWA");
        while (TravNode != null) {
            System.out.printf(red + "%-10d%-30s%-30s%-30s%.2f\n\n" + rs, ctr, TravNode.studno, TravNode.studname, TravNode.studcrsyr, TravNode.GWA);
            TravNode = TravNode.next;
            ctr++;
        }
    }
    System.out.println(yellow + "\n--------------------------------------------------------------------------------------------------------------\n" + rs);
}
```

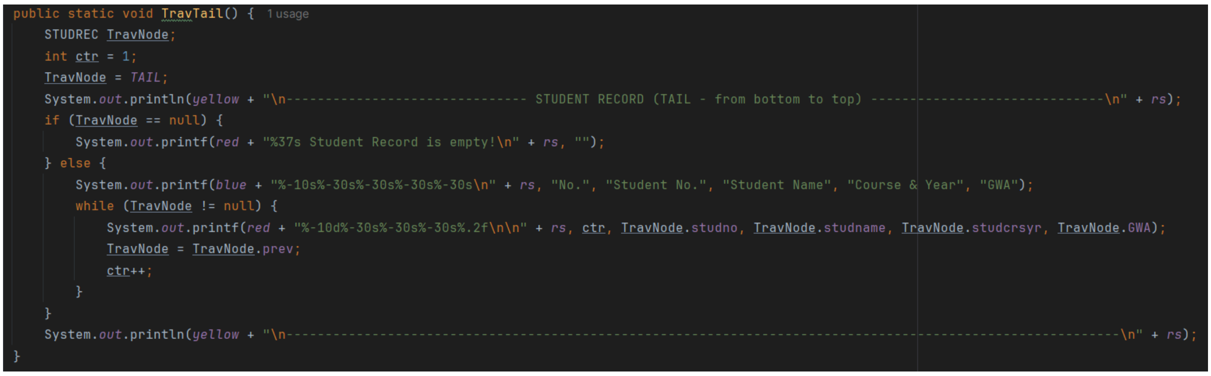
Output



```
------------------------------ STUDENT RECORD (HEAD - from top to bottom) ------------------------------

No.       Student No.                   Student Name                  Course & Year                 GWA
1         1001                          Gojo Satoru                   Computer Science - 1          1.00

2         1002                          Toji Fushiguro                Computer Science - 2          3.00



--------------------------------------------------------------------------------------------------------------------


--------------------------------------------------------------------------------------------------------------------

Press Enter to return to main menu...

--------------------------------------------------------------------------------------------------------------------
```

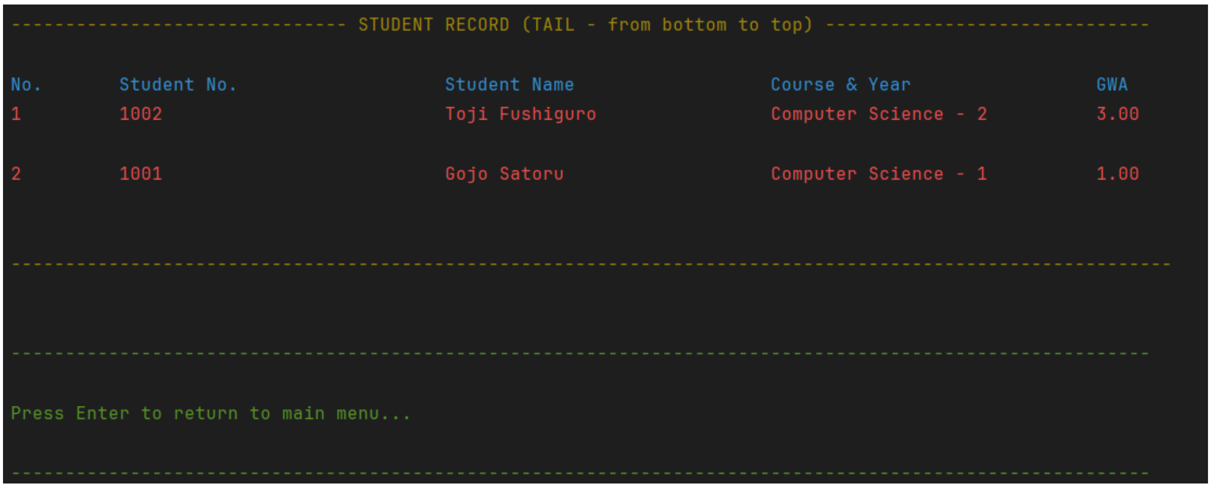
IV. Traversal of Double/Doubly Linked List (Tail - Bottom to Top)

Source Code in Java



```java
public static void TravTail() {  1 usage
    STUDREC TravNode;
    int ctr = 1;
    TravNode = TAIL;
    System.out.println(yellow + "\n------------------------------ STUDENT RECORD (TAIL - from bottom to top) ------------------------------\n" + rs);
    if (TravNode == null) {
        System.out.printf(red + "%37s Student Record is empty!\n" + rs, "");
    } else {
        System.out.printf(blue + "%-10s%-30s%-30s%-30s%-30s\n" + rs, "No.", "Student No.", "Student Name", "Course & Year", "GWA");
        while (TravNode != null) {
            System.out.printf(red + "%-10d%-30s%-30s%-30s%.2f\n\n" + rs, ctr, TravNode.studno, TravNode.studname, TravNode.studcrsyr, TravNode.GWA);
            TravNode = TravNode.prev;
            ctr++;
        }
    }
    System.out.println(yellow + "\n----------------------------------------------------------------------------------------------------------\n" + rs);
}
```

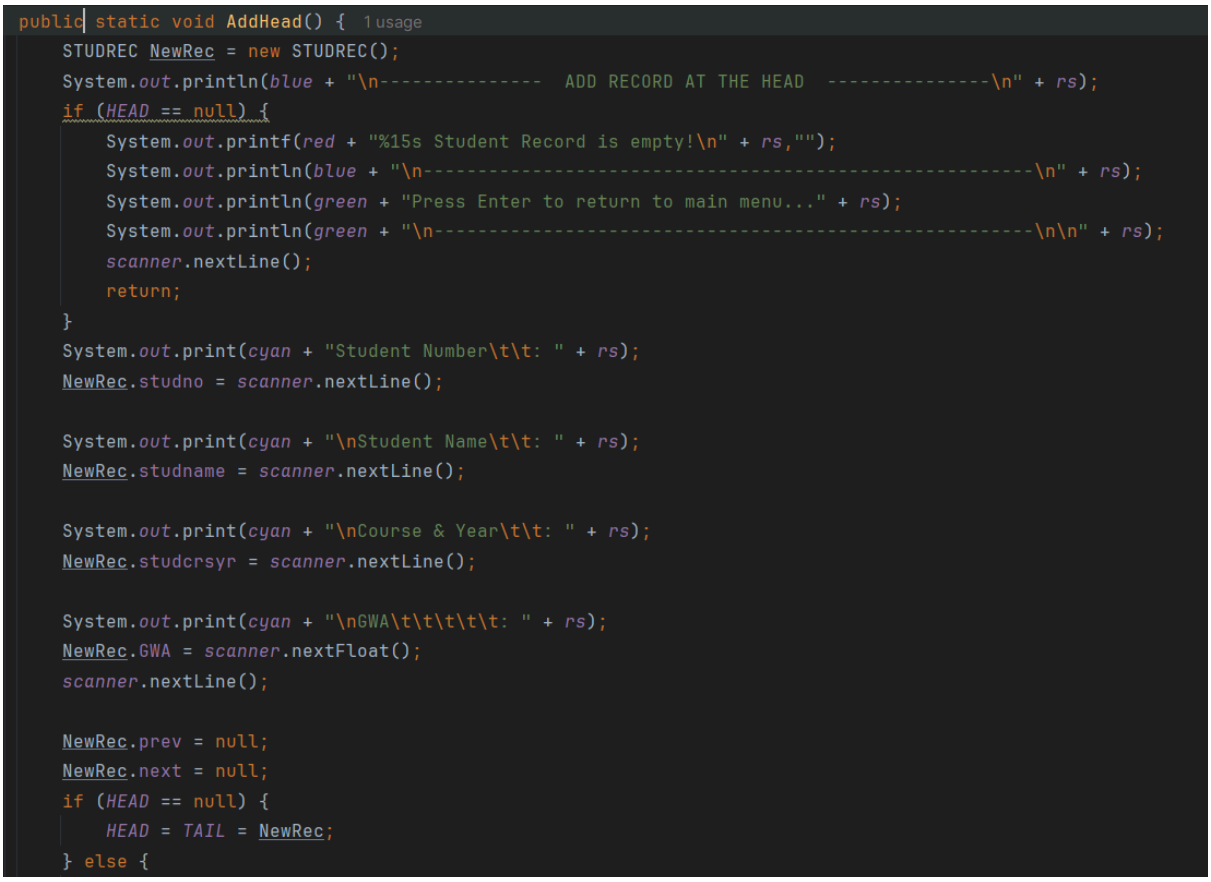
Output



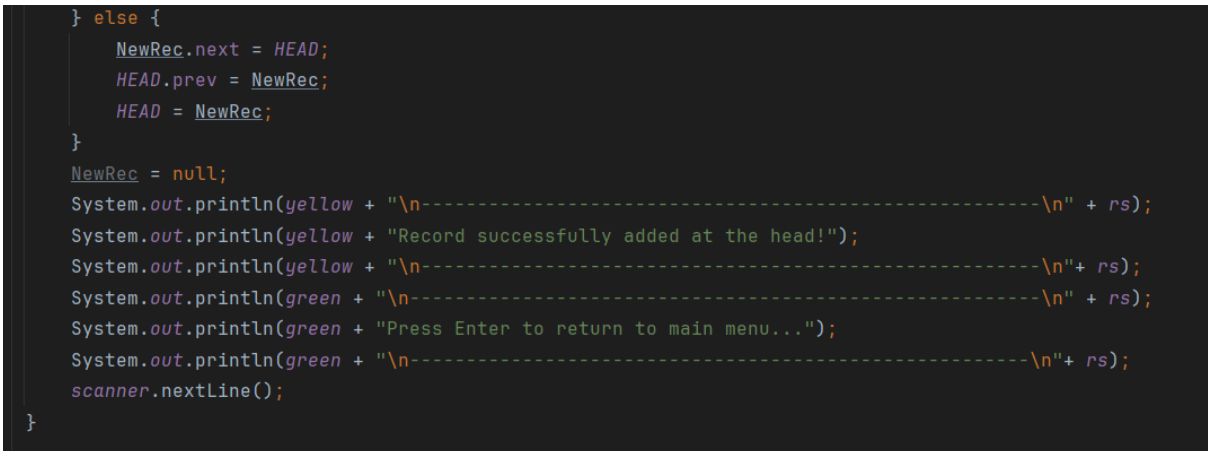
```
------------------------------ STUDENT RECORD (TAIL - from bottom to top) ------------------------------

No.       Student No.                   Student Name                  Course & Year                 GWA
1         1002                          Toji Fushiguro                Computer Science - 2          3.00

2         1001                          Gojo Satoru                   Computer Science - 1          1.00


-----------------------------------------------------------------------------------------------------------



----------------------------------------------------------------------------------------------------------


Press Enter to return to main menu...


----------------------------------------------------------------------------------------------------------
```

V. Adding/Insertion of Node in Double/Doubly Linked List (at the Head)

Source Code in Java

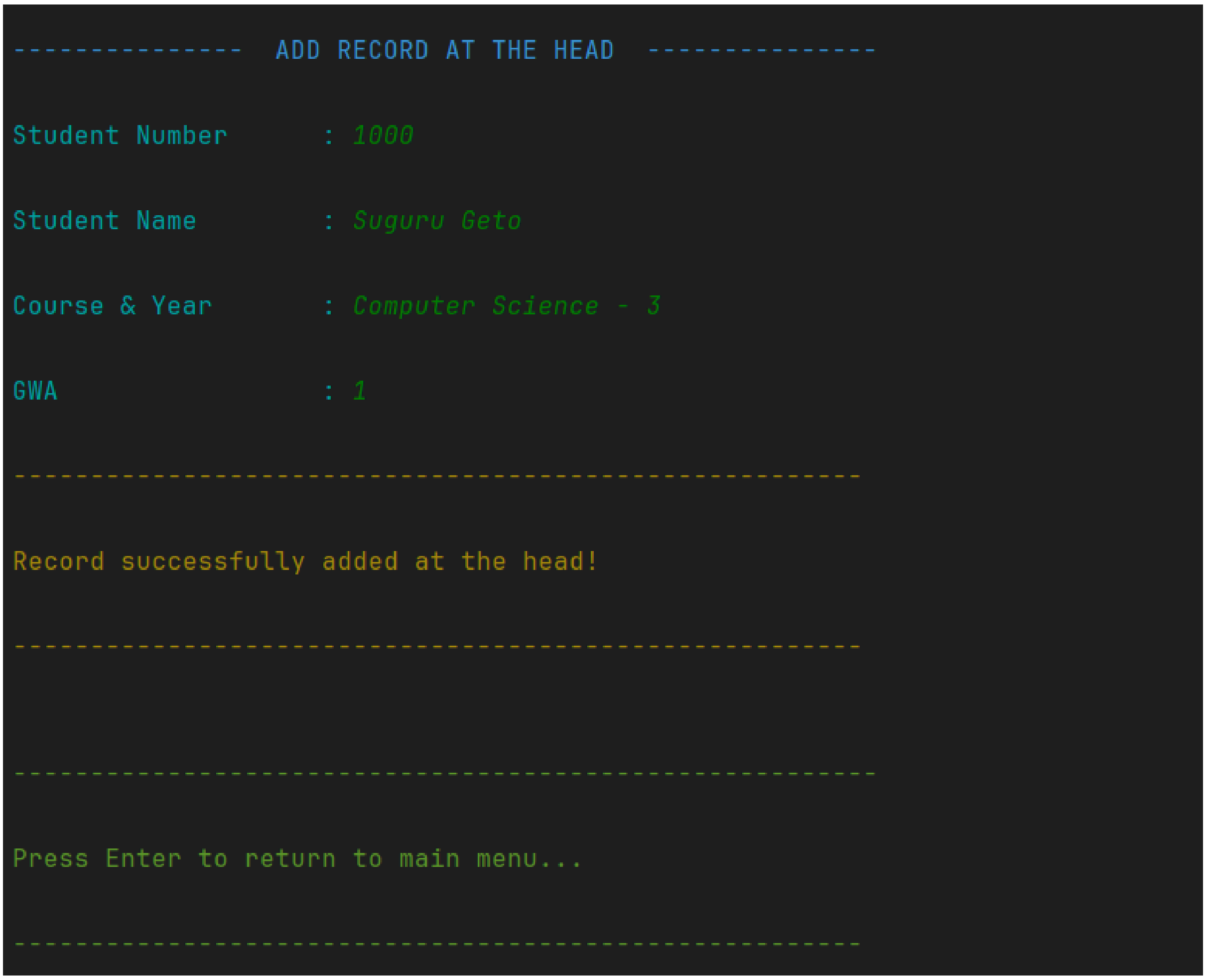


```
public static void AddHead() {  1 usage
    STUDREC NewRec = new STUDREC();
    System.out.println(blue + "\n--------------  ADD RECORD AT THE HEAD  --------------\n" + rs);
    if (HEAD == null) {
        System.out.printf(red + "%15s Student Record is empty!\n" + rs,"");
        System.out.println(blue + "\n--------------------------------------------------------\n" + rs);
        System.out.println(green + "Press Enter to return to main menu..." + rs);
        System.out.println(green + "\n--------------------------------------------------------\n\n" + rs);
        scanner.nextLine();
        return;
    }
    System.out.print(cyan + "Student Number\t\t: " + rs);
    NewRec.studno = scanner.nextLine();

    System.out.print(cyan + "\nStudent Name\t\t: " + rs);
    NewRec.studname = scanner.nextLine();

    System.out.print(cyan + "\nCourse & Year\t\t: " + rs);
    NewRec.studcrsyr = scanner.nextLine();

    System.out.print(cyan + "\nGWA\t\t\t\t\t: " + rs);
    NewRec.GWA = scanner.nextFloat();
    scanner.nextLine();

    NewRec.prev = null;
    NewRec.next = null;
    if (HEAD == null) {
        HEAD = TAIL = NewRec;
    } else {
```

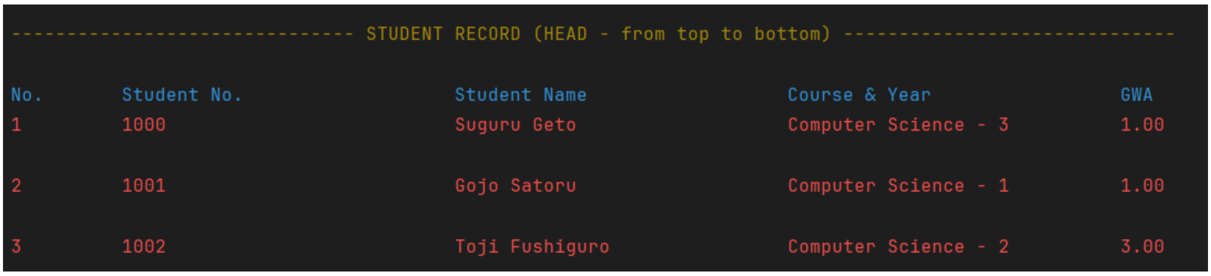


```
    } else {
        NewRec.next = HEAD;
        HEAD.prev = NewRec;
        HEAD = NewRec;
    }
    NewRec = null;
    System.out.println(yellow + "\n--------------------------------------------------------\n" + rs);
    System.out.println(yellow + "Record successfully added at the head!");
    System.out.println(yellow + "\n--------------------------------------------------------\n"+ rs);
    System.out.println(green + "\n--------------------------------------------------------\n" + rs);
    System.out.println(green + "Press Enter to return to main menu...");
    System.out.println(green + "\n--------------------------------------------------------\n"+ rs);
    scanner.nextLine();
}
```

Output



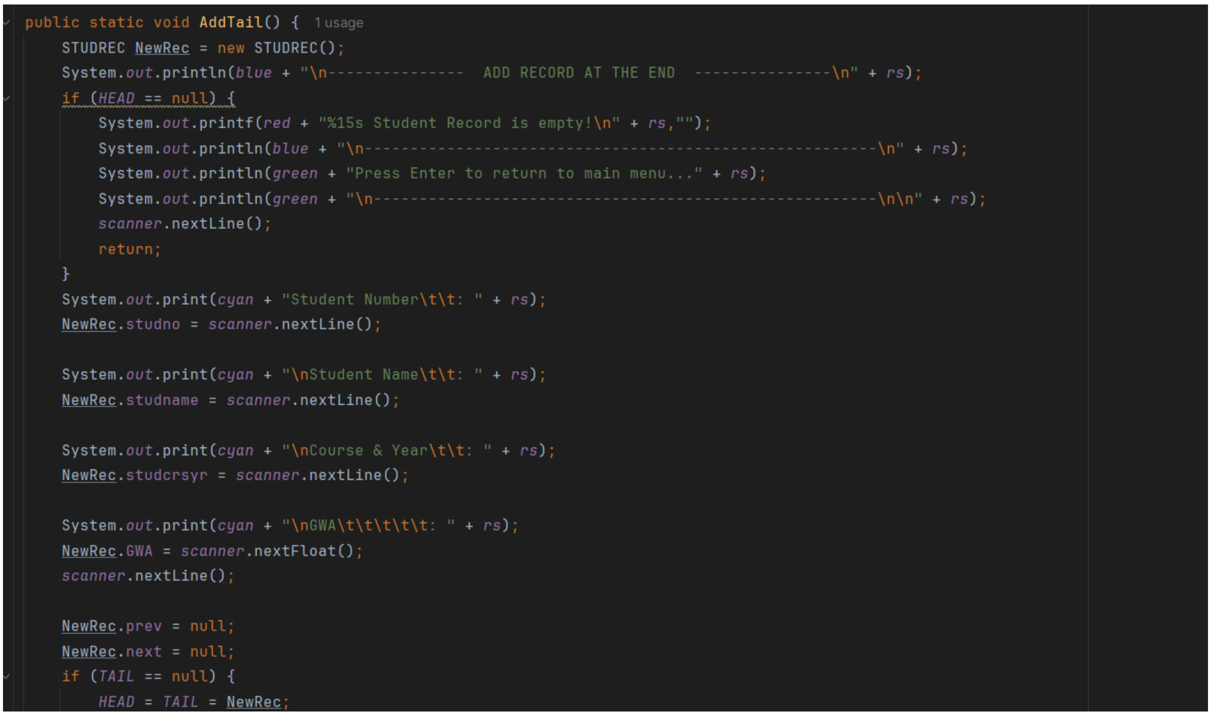
```
---------------  ADD RECORD AT THE HEAD  ---------------

Student Number      : 1000

Student Name        : Suguru Geto

Course & Year       : Computer Science - 3

GWA                 : 1

-------------------------------------------------------

Record successfully added at the head!

-------------------------------------------------------


--------------------------------------------------------

Press Enter to return to main menu...

-------------------------------------------------------
```



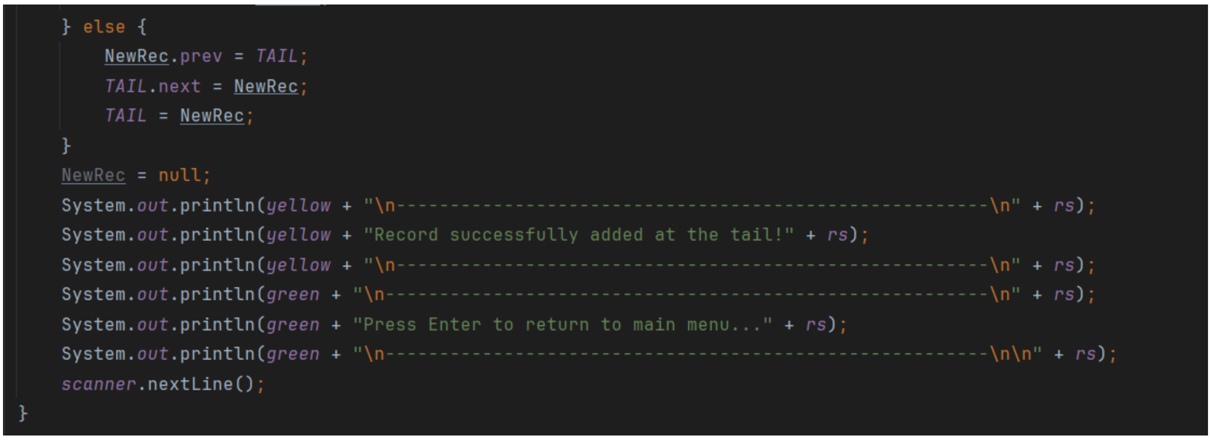
```
------------------------------- STUDENT RECORD (HEAD - from top to bottom) ------------------------------

No.       Student No.                   Student Name                  Course & Year                 GWA
1         1000                          Suguru Geto                   Computer Science - 3          1.00

2         1001                          Gojo Satoru                   Computer Science - 1          1.00

3         1002                          Toji Fushiguro                Computer Science - 2          3.00
```

VI. Adding/Insertion of Node in Double/Doubly Linked List (at the Tail)

Source Code in Java

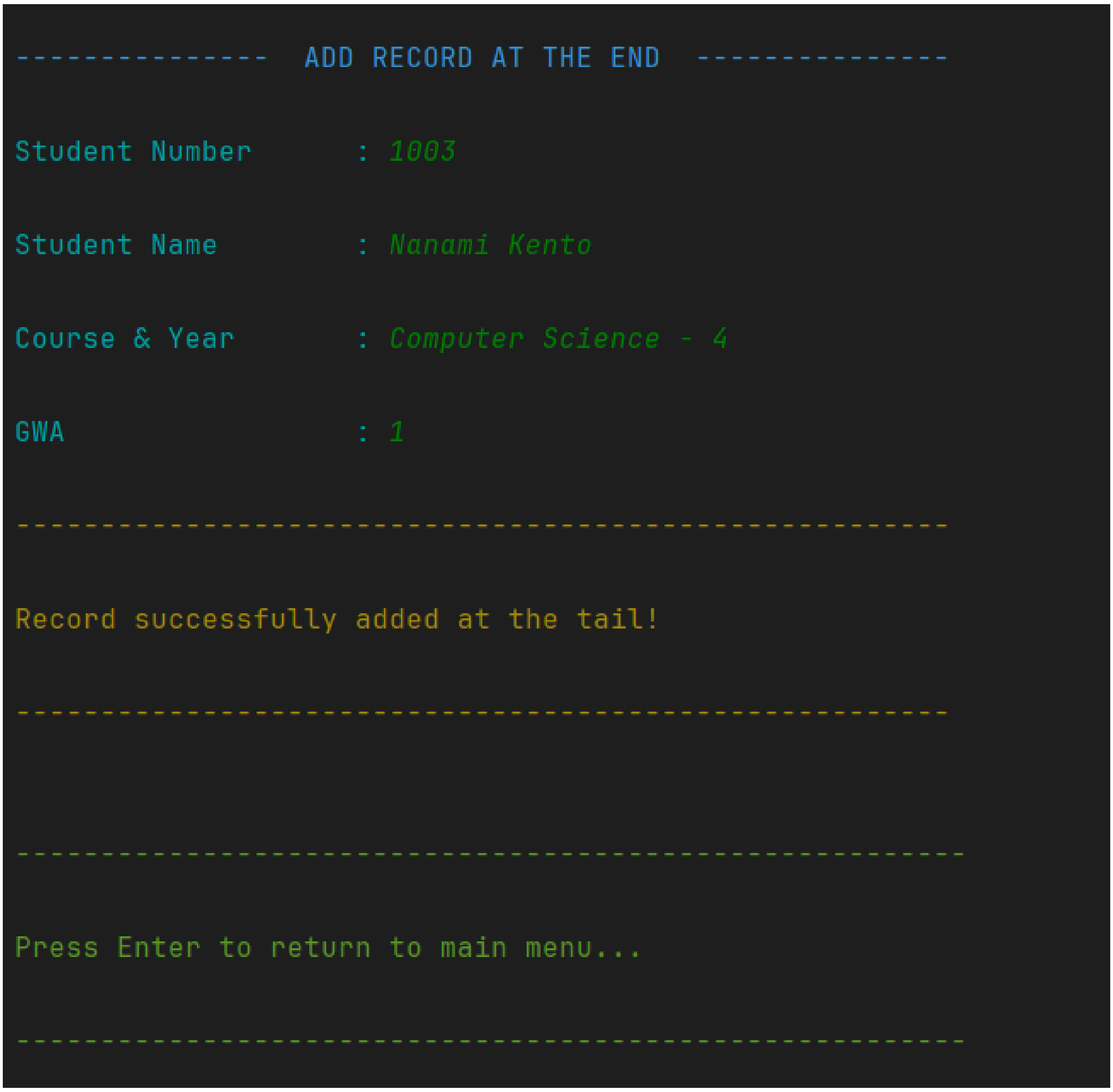


```java
public static void AddTail() {  1 usage
    STUDREC NewRec = new STUDREC();
    System.out.println(blue + "\n---------------  ADD RECORD AT THE END  ---------------\n" + rs);
    if (HEAD == null) {
        System.out.printf(red + "%15s Student Record is empty!\n" + rs,"");
        System.out.println(blue + "\n--------------------------------------------------------\n" + rs);
        System.out.println(green + "Press Enter to return to main menu..." + rs);
        System.out.println(green + "\n--------------------------------------------------------\n\n" + rs);
        scanner.nextLine();
        return;
    }
    System.out.print(cyan + "Student Number\t\t: " + rs);
    NewRec.studno = scanner.nextLine();

    System.out.print(cyan + "\nStudent Name\t\t: " + rs);
    NewRec.studname = scanner.nextLine();

    System.out.print(cyan + "\nCourse & Year\t\t: " + rs);
    NewRec.studcrsyr = scanner.nextLine();

    System.out.print(cyan + "\nGWA\t\t\t\t\t: " + rs);
    NewRec.GWA = scanner.nextFloat();
    scanner.nextLine();

    NewRec.prev = null;
    NewRec.next = null;
    if (TAIL == null) {
        HEAD = TAIL = NewRec;
```

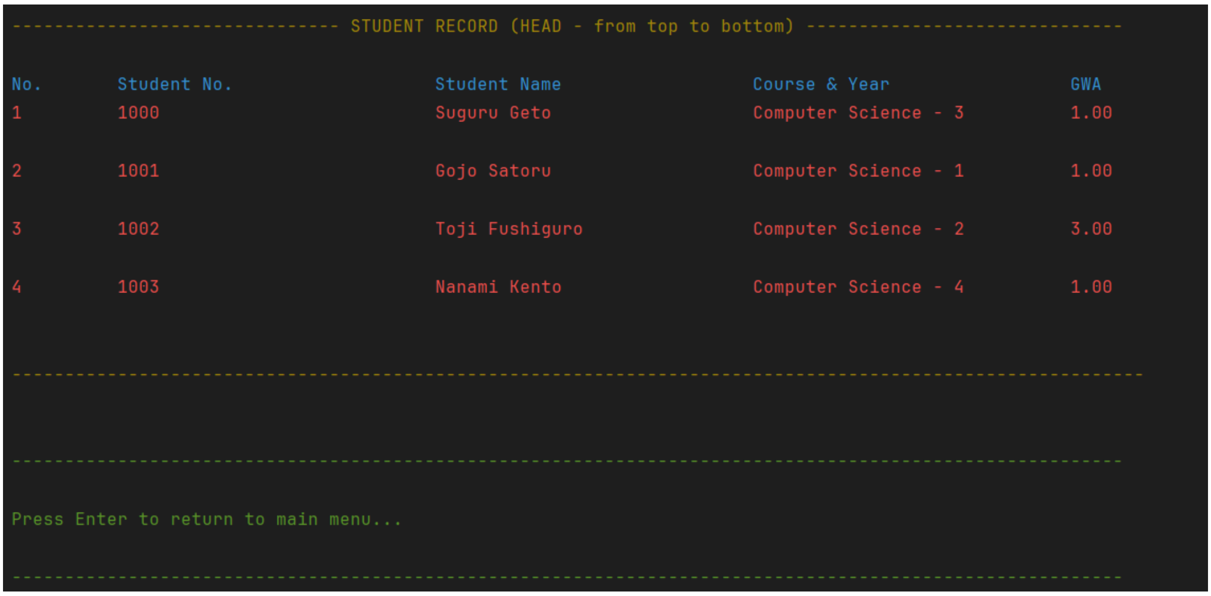


```java
    } else {
        NewRec.prev = TAIL;
        TAIL.next = NewRec;
        TAIL = NewRec;
    }
    NewRec = null;
    System.out.println(yellow + "\n--------------------------------------------------------\n" + rs);
    System.out.println(yellow + "Record successfully added at the tail!" + rs);
    System.out.println(yellow + "\n--------------------------------------------------------\n" + rs);
    System.out.println(green + "\n--------------------------------------------------------\n" + rs);
    System.out.println(green + "Press Enter to return to main menu..." + rs);
    System.out.println(green + "\n--------------------------------------------------------\n\n" + rs);
    scanner.nextLine();
}
```

Output



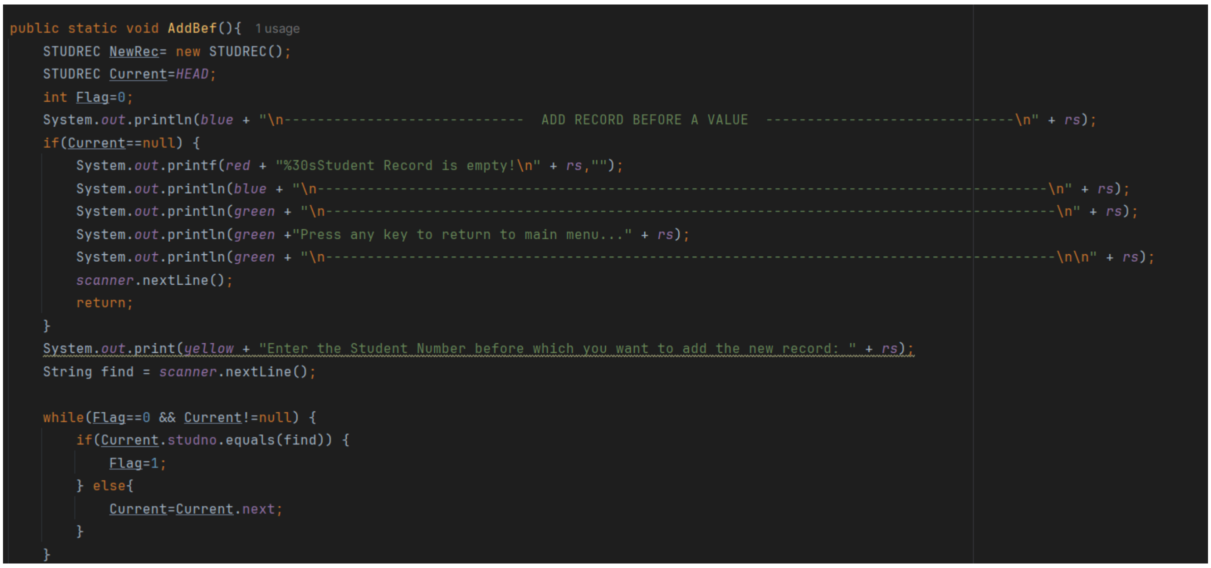
```
---------------  ADD RECORD AT THE END  ---------------

Student Number      : 1003

Student Name        : Nanami Kento

Course & Year       : Computer Science - 4

GWA                 : 1

-------------------------------------------------------

Record successfully added at the tail!

-------------------------------------------------------


--------------------------------------------------------

Press Enter to return to main menu...

--------------------------------------------------------
```



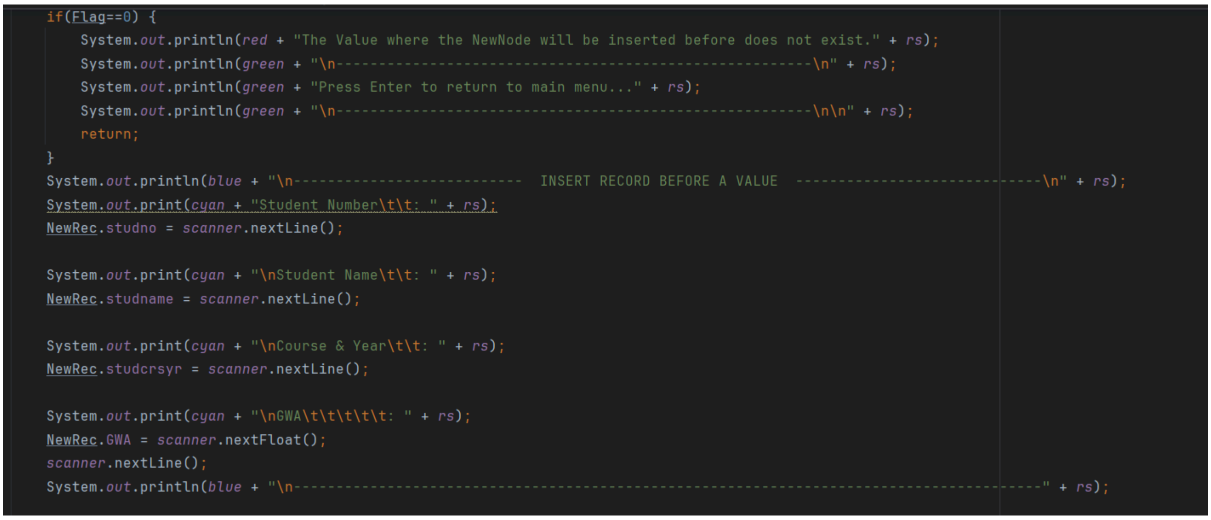
```
------------------------------- STUDENT RECORD (HEAD - from top to bottom) ------------------------------

No.       Student No.                   Student Name                  Course & Year                 GWA
1         1000                          Suguru Geto                   Computer Science - 3          1.00

2         1001                          Gojo Satoru                   Computer Science - 1          1.00

3         1002                          Toji Fushiguro                Computer Science - 2          3.00

4         1003                          Nanami Kento                  Computer Science - 4          1.00


-----------------------------------------------------------------------------------------------------------


---------------------------------------------------------------------------------------------------------

Press Enter to return to main menu...

---------------------------------------------------------------------------------------------------------
```

VII. Adding/Insertion of Node in Double/Doubly Linked List (Before a Value/Data)

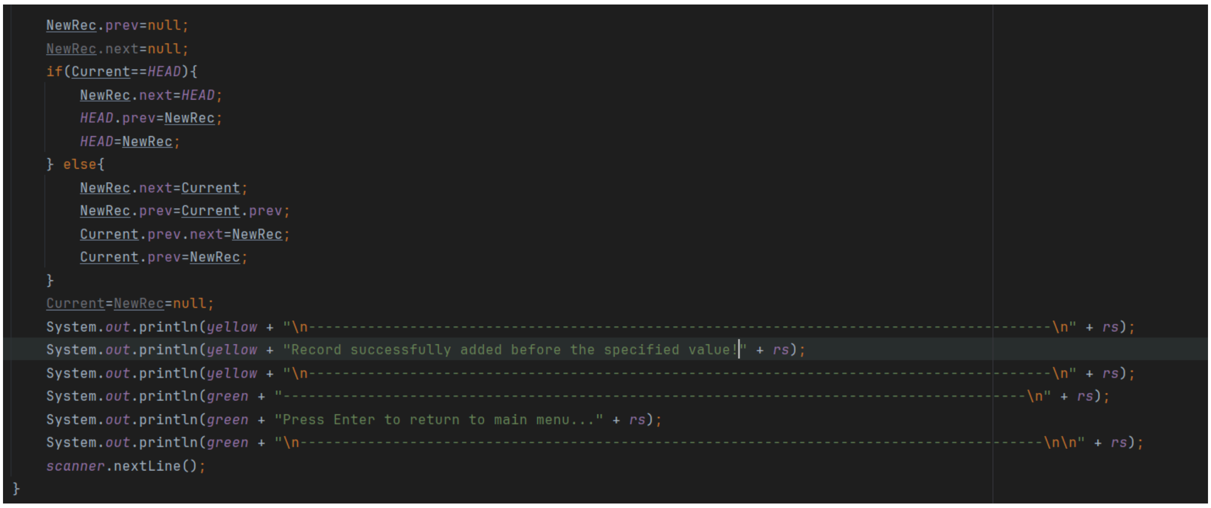
Source Code in Java



```java
public static void AddBef(){  1 usage
    STUDREC NewRec= new STUDREC();
    STUDREC Current=HEAD;
    int Flag=0;
    System.out.println(blue + "\n-----------------------------  ADD RECORD BEFORE A VALUE  ------------------------------\n" + rs);
    if(Current==null) {
        System.out.printf(red + "%30sStudent Record is empty!\n" + rs,"");
        System.out.println(blue + "\n-----------------------------------------------------------------------------------------\n" + rs);
        System.out.println(green + "\n----------------------------------------------------------------------------------------\n" + rs);
        System.out.println(green +"Press any key to return to main menu..." + rs);
        System.out.println(green + "\n----------------------------------------------------------------------------------------\n\n" + rs);
        scanner.nextLine();
        return;
    }
    System.out.print(yellow + "Enter the Student Number before which you want to add the new record: " + rs);
    String find = scanner.nextLine();

    while(Flag==0 && Current!=null) {
        if(Current.studno.equals(find)) {
            Flag=1;
        } else{
            Current=Current.next;
        }
    }
```

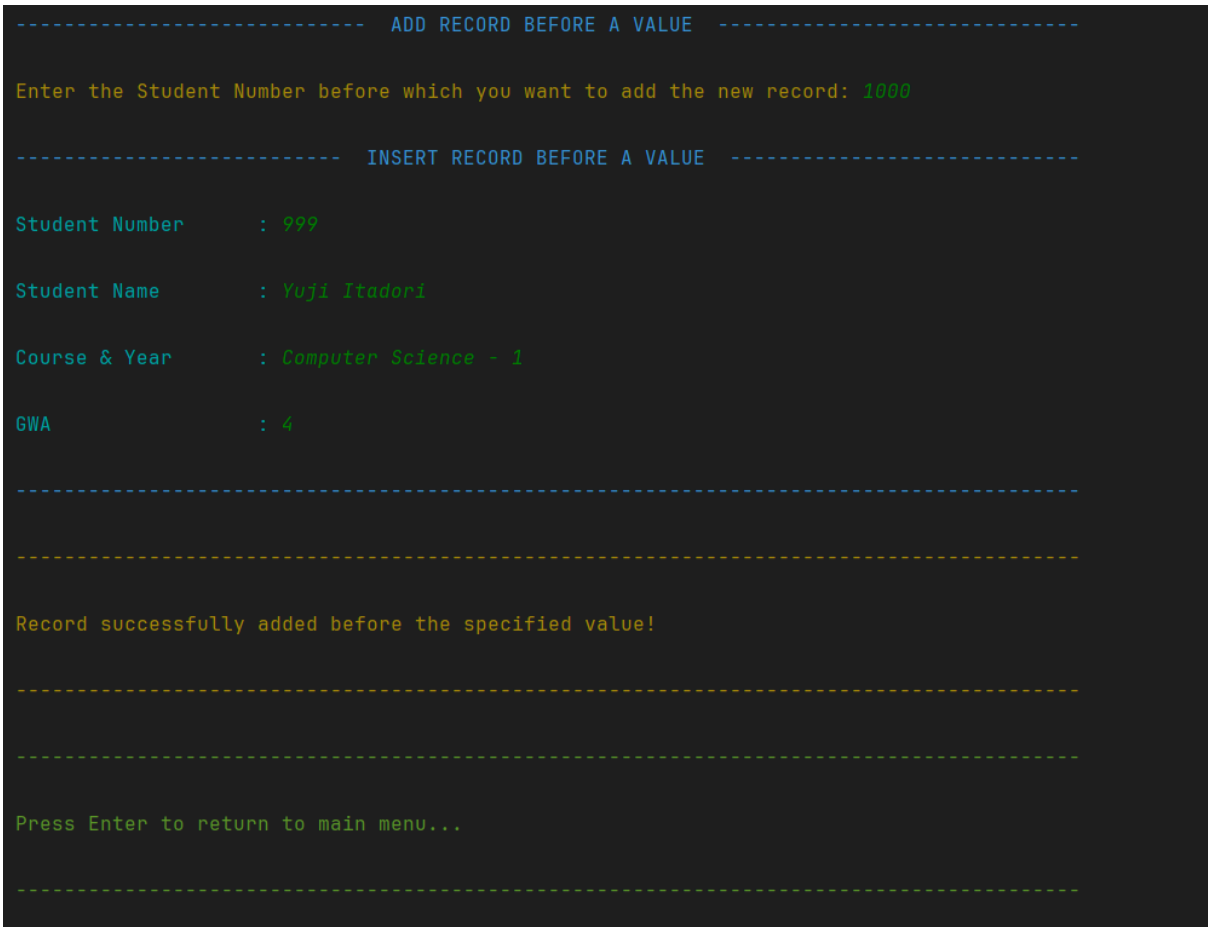


```java
    if(Flag==0) {
        System.out.println(red + "The Value where the NewNode will be inserted before does not exist." + rs);
        System.out.println(green + "\n----------------------------------------------------------\n" + rs);
        System.out.println(green + "Press Enter to return to main menu..." + rs);
        System.out.println(green + "\n----------------------------------------------------------\n\n" + rs);
        return;
    }
    System.out.println(blue + "\n---------------------------  INSERT RECORD BEFORE A VALUE  ------------------------------\n" + rs);
    System.out.print(cyan + "Student Number\t\t: " + rs);
    NewRec.studno = scanner.nextLine();

    System.out.print(cyan + "\nStudent Name\t\t: " + rs);
    NewRec.studname = scanner.nextLine();

    System.out.print(cyan + "\nCourse & Year\t\t: " + rs);
    NewRec.studcrsyr = scanner.nextLine();

    System.out.print(cyan + "\nGWA\t\t\t\t\t: " + rs);
    NewRec.GWA = scanner.nextFloat();
    scanner.nextLine();
    System.out.println(blue + "\n-------------------------------------------------------------------------------------------" + rs);
```

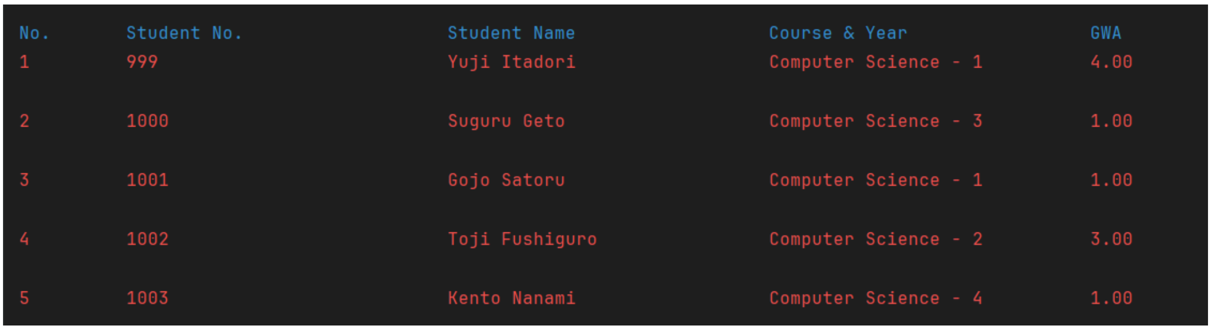


```java
    NewRec.prev=null;
    NewRec.next=null;
    if(Current==HEAD){
        NewRec.next=HEAD;
        HEAD.prev=NewRec;
        HEAD=NewRec;
    } else{
        NewRec.next=Current;
        NewRec.prev=Current.prev;
        Current.prev.next=NewRec;
        Current.prev=NewRec;
    }
    Current=NewRec=null;
    System.out.println(yellow + "\n-----------------------------------------------------------------------------------------\n" + rs);
    System.out.println(yellow + "Record successfully added before the specified value!" + rs);
    System.out.println(yellow + "\n-----------------------------------------------------------------------------------------\n" + rs);
    System.out.println(green + "------------------------------------------------------------------------------------------\n" + rs);
    System.out.println(green + "Press Enter to return to main menu..." + rs);
    System.out.println(green + "\n-------------------------------------------------------------------------------------------\n\n" + rs);
    scanner.nextLine();
}
```

Output



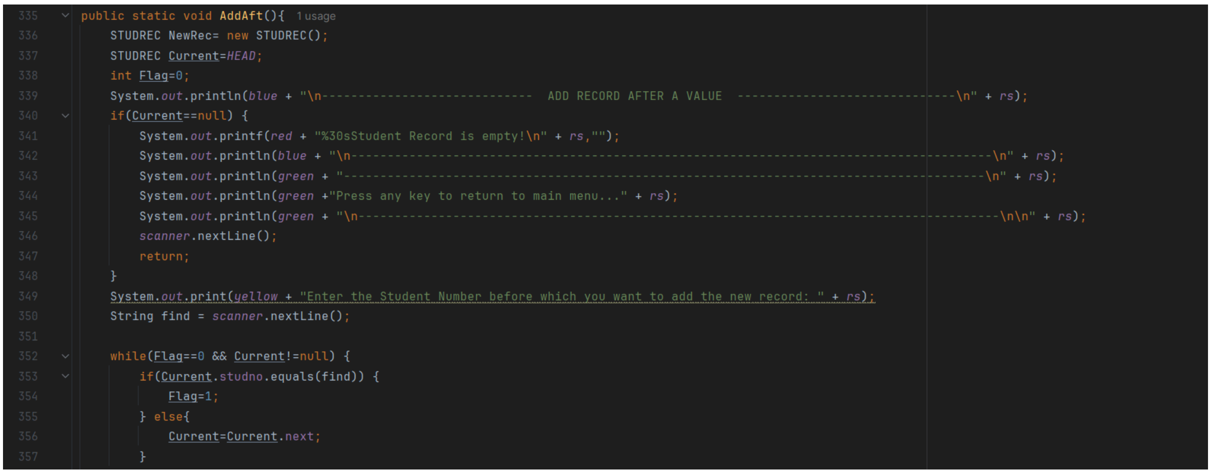
```
-----------------------------  ADD RECORD BEFORE A VALUE  ------------------------------

Enter the Student Number before which you want to add the new record: 1000

---------------------------  INSERT RECORD BEFORE A VALUE  -----------------------------

Student Number      : 999

Student Name        : Yuji Itadori

Course & Year       : Computer Science - 1

GWA                 : 4

----------------------------------------------------------------------------------------

----------------------------------------------------------------------------------------

Record successfully added before the specified value!

----------------------------------------------------------------------------------------

----------------------------------------------------------------------------------------

Press Enter to return to main menu...

----------------------------------------------------------------------------------------
```



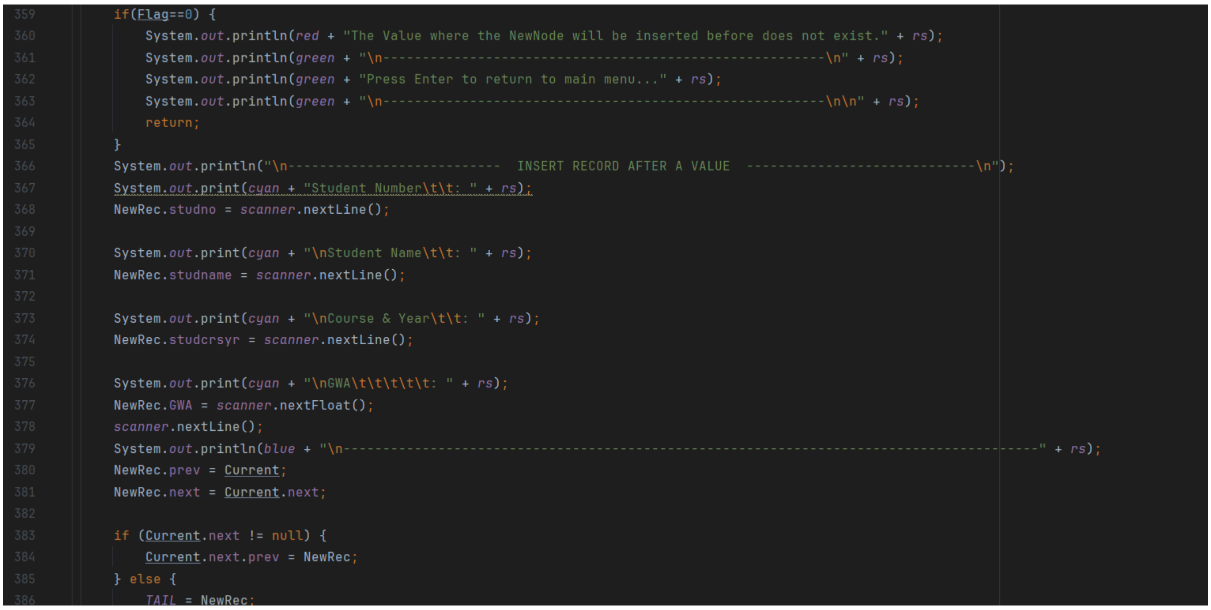
```
No.      Student No.                Student Name               Course & Year              GWA
1        999                        Yuji Itadori               Computer Science - 1       4.00

2        1000                       Suguru Geto                Computer Science - 3       1.00

3        1001                       Gojo Satoru                Computer Science - 1       1.00

4        1002                       Toji Fushiguro             Computer Science - 2       3.00

5        1003                       Kento Nanami               Computer Science - 4       1.00
```

VIII. Adding/Insertion of Node in Double/Doubly Linked List (After a Value/Data)

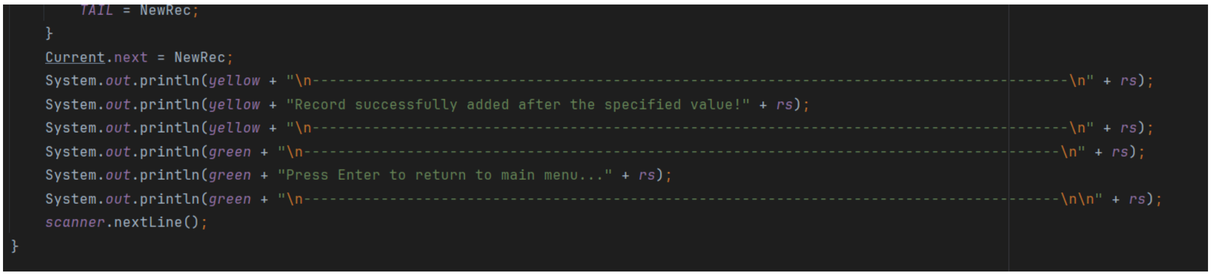
Source Code in Java



```java
public static void AddAft(){  1 usage
    STUDREC NewRec= new STUDREC();
    STUDREC Current=HEAD;
    int Flag=0;
    System.out.println(blue + "\n-----------------------------  ADD RECORD AFTER A VALUE  ------------------------------\n" + rs);
    if(Current==null) {
        System.out.printf(red + "%30sStudent Record is empty!\n" + rs,"");
        System.out.println(blue + "\n-----------------------------------------------------------------------------------------\n" + rs);
        System.out.println(green + "----------------------------------------------------------------------------------------\n" + rs);
        System.out.println(green +"Press any key to return to main menu..." + rs);
        System.out.println(green + "\n-----------------------------------------------------------------------------------------\n\n" + rs);
        scanner.nextLine();
        return;
    }
    System.out.print(yellow + "Enter the Student Number before which you want to add the new record: " + rs);
    String find = scanner.nextLine();

    while(Flag==0 && Current!=null) {
        if(Current.studno.equals(find)) {
            Flag=1;
        } else{
            Current=Current.next;
        }
```

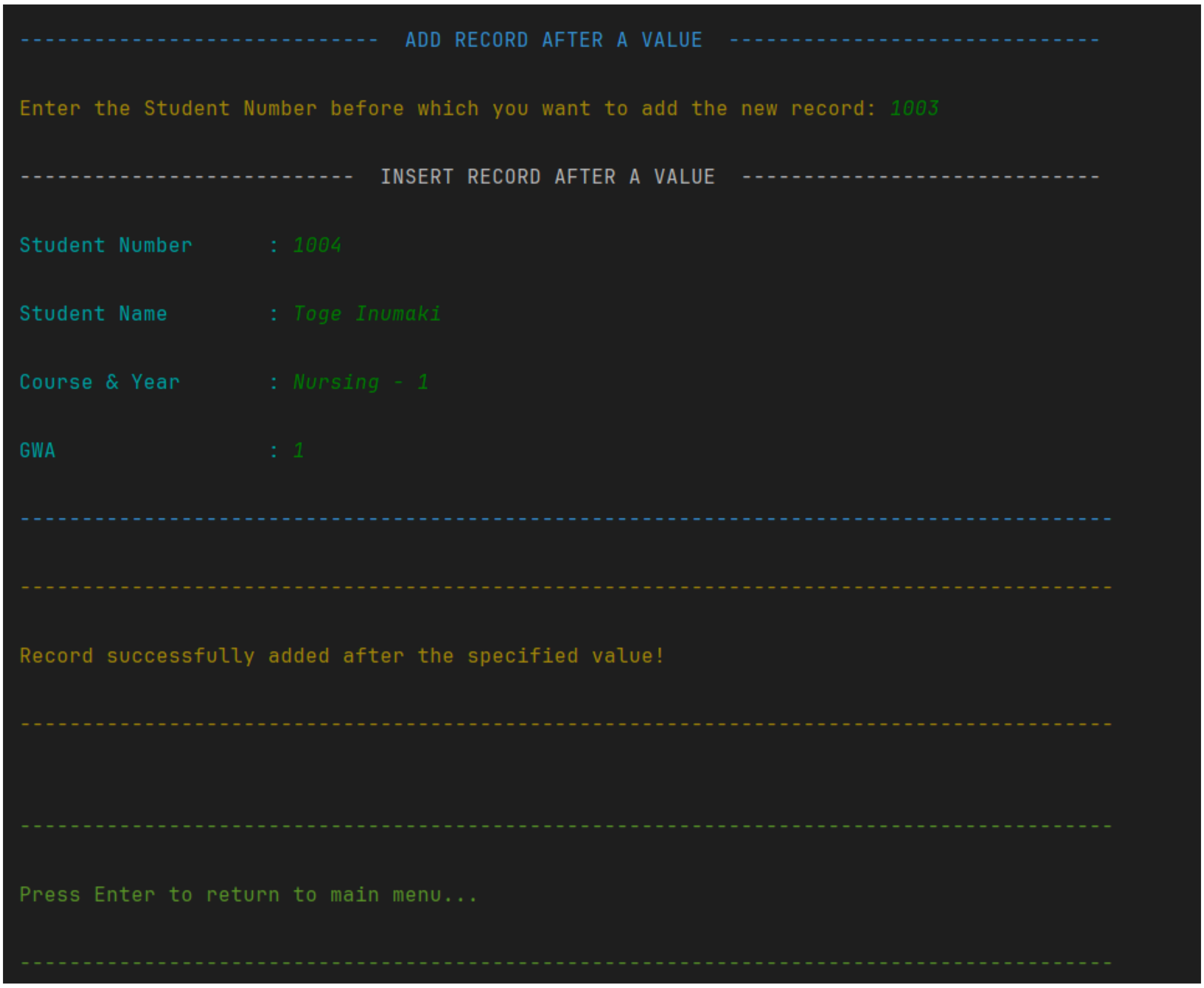


```java
    if(Flag==0) {
        System.out.println(red + "The Value where the NewNode will be inserted before does not exist." + rs);
        System.out.println(green + "\n----------------------------------------------------------\n" + rs);
        System.out.println(green + "Press Enter to return to main menu..." + rs);
        System.out.println(green + "\n----------------------------------------------------------\n\n" + rs);
        return;
    }
    System.out.println("\n---------------------------  INSERT RECORD AFTER A VALUE  ------------------------------\n");
    System.out.print(cyan + "Student Number\t\t: " + rs);
    NewRec.studno = scanner.nextLine();

    System.out.print(cyan + "\nStudent Name\t\t: " + rs);
    NewRec.studname = scanner.nextLine();

    System.out.print(cyan + "\nCourse & Year\t\t: " + rs);
    NewRec.studcrsyr = scanner.nextLine();

    System.out.print(cyan + "\nGWA\t\t\t\t\t: " + rs);
    NewRec.GWA = scanner.nextFloat();
    scanner.nextLine();
    System.out.println(blue + "\n-------------------------------------------------------------------------------------------" + rs);
    NewRec.prev = Current;
    NewRec.next = Current.next;

    if (Current.next != null) {
        Current.next.prev = NewRec;
    } else {
        TAIL = NewRec;
```

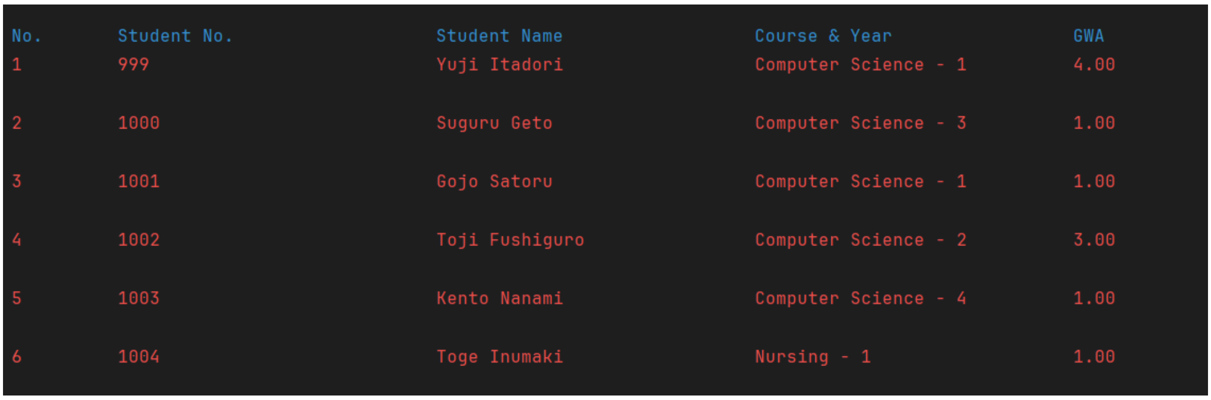


```java
        TAIL = NewRec;
    }
    Current.next = NewRec;
    System.out.println(yellow + "\n-----------------------------------------------------------------\n" + rs);
    System.out.println(yellow + "Record successfully added after the specified value!" + rs);
    System.out.println(yellow + "\n-----------------------------------------------------------------\n" + rs);
    System.out.println(green + "\n-----------------------------------------------------------------\n" + rs);
    System.out.println(green + "Press Enter to return to main menu..." + rs);
    System.out.println(green + "\n-----------------------------------------------------------------\n\n" + rs);
    scanner.nextLine();
}
```

Output



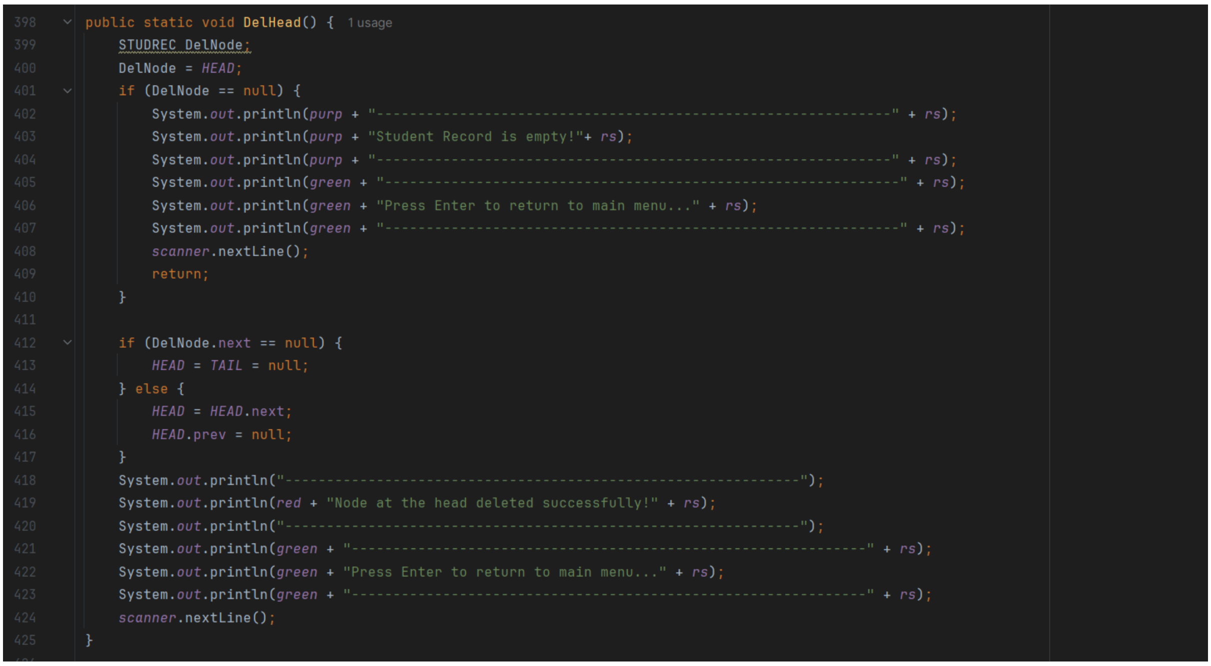
```
-----------------------------  ADD RECORD AFTER A VALUE  ------------------------------

Enter the Student Number before which you want to add the new record: 1003

---------------------------  INSERT RECORD AFTER A VALUE  -----------------------------

Student Number      : 1004

Student Name        : Toge Inumaki

Course & Year       : Nursing - 1

GWA                 : 1

---------------------------------------------------------------------------------------


---------------------------------------------------------------------------------------

Record successfully added after the specified value!

---------------------------------------------------------------------------------------



---------------------------------------------------------------------------------------

Press Enter to return to main menu...

---------------------------------------------------------------------------------------
```



```
No.      Student No.              Student Name             Course & Year            GWA
1        999                      Yuji Itadori             Computer Science - 1     4.00

2        1000                     Suguru Geto              Computer Science - 3     1.00

3        1001                     Gojo Satoru              Computer Science - 1     1.00

4        1002                     Toji Fushiguro           Computer Science - 2     3.00

5        1003                     Kento Nanami             Computer Science - 4     1.00

6        1004                     Toge Inumaki             Nursing - 1              1.00
```

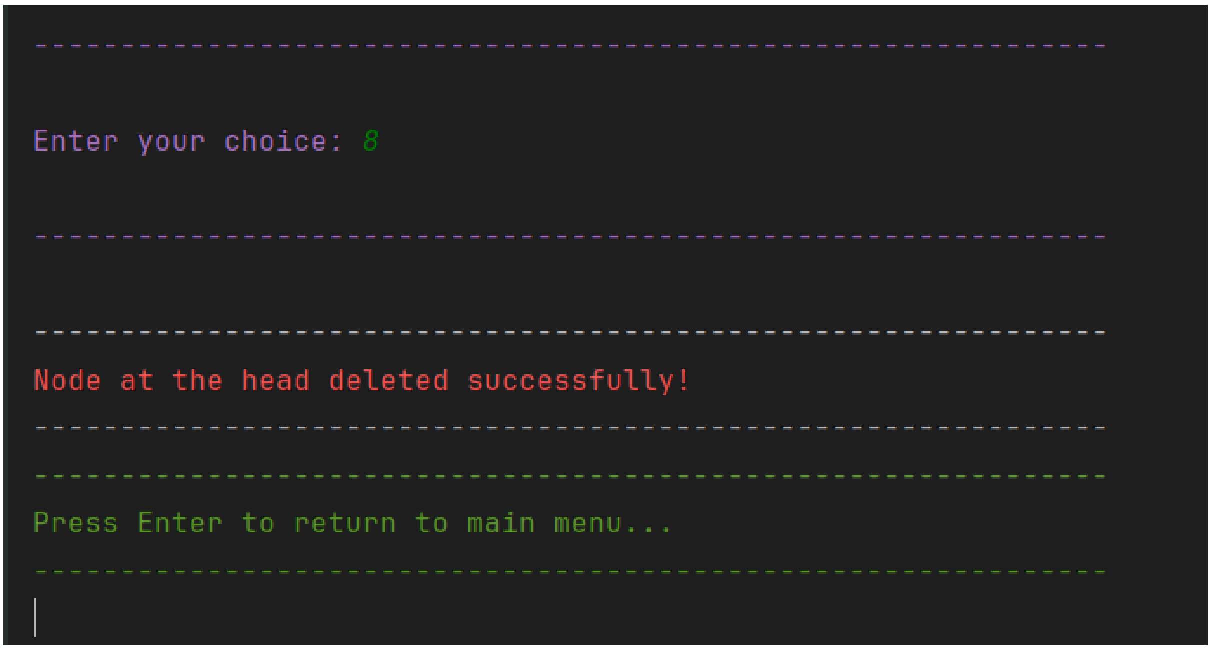
IX. Deletion of Node in Double/Doubly Linked List (at the Head)

Source Code in Java

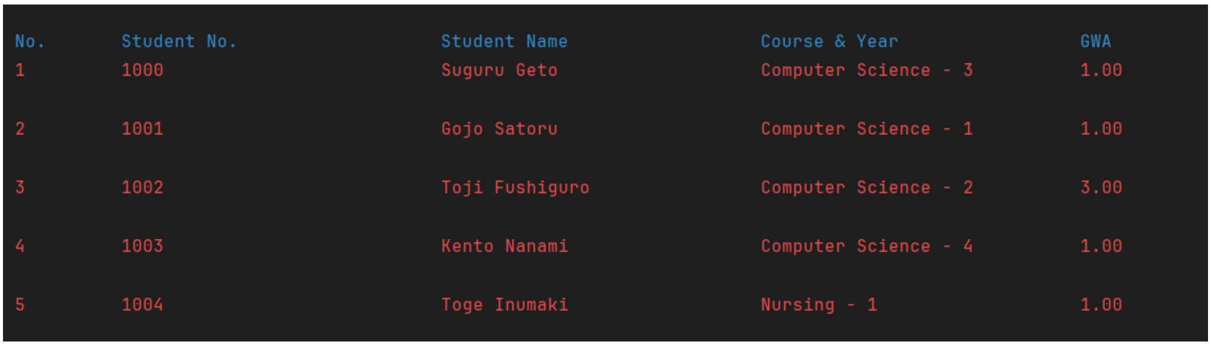


```java
public static void DelHead() {  1 usage
    STUDREC DelNode;
    DelNode = HEAD;
    if (DelNode == null) {
        System.out.println(purp + "--------------------------------------------------------------" + rs);
        System.out.println(purp + "Student Record is empty!"+ rs);
        System.out.println(purp + "--------------------------------------------------------------" + rs);
        System.out.println(green + "--------------------------------------------------------------" + rs);
        System.out.println(green + "Press Enter to return to main menu..." + rs);
        System.out.println(green + "--------------------------------------------------------------" + rs);
        scanner.nextLine();
        return;
    }

    if (DelNode.next == null) {
        HEAD = TAIL = null;
    } else {
        HEAD = HEAD.next;
        HEAD.prev = null;
    }
    System.out.println("--------------------------------------------------------------");
    System.out.println(red + "Node at the head deleted successfully!" + rs);
    System.out.println("--------------------------------------------------------------");
    System.out.println(green + "--------------------------------------------------------------" + rs);
    System.out.println(green + "Press Enter to return to main menu..." + rs);
    System.out.println(green + "--------------------------------------------------------------" + rs);
    scanner.nextLine();
}
```

Output



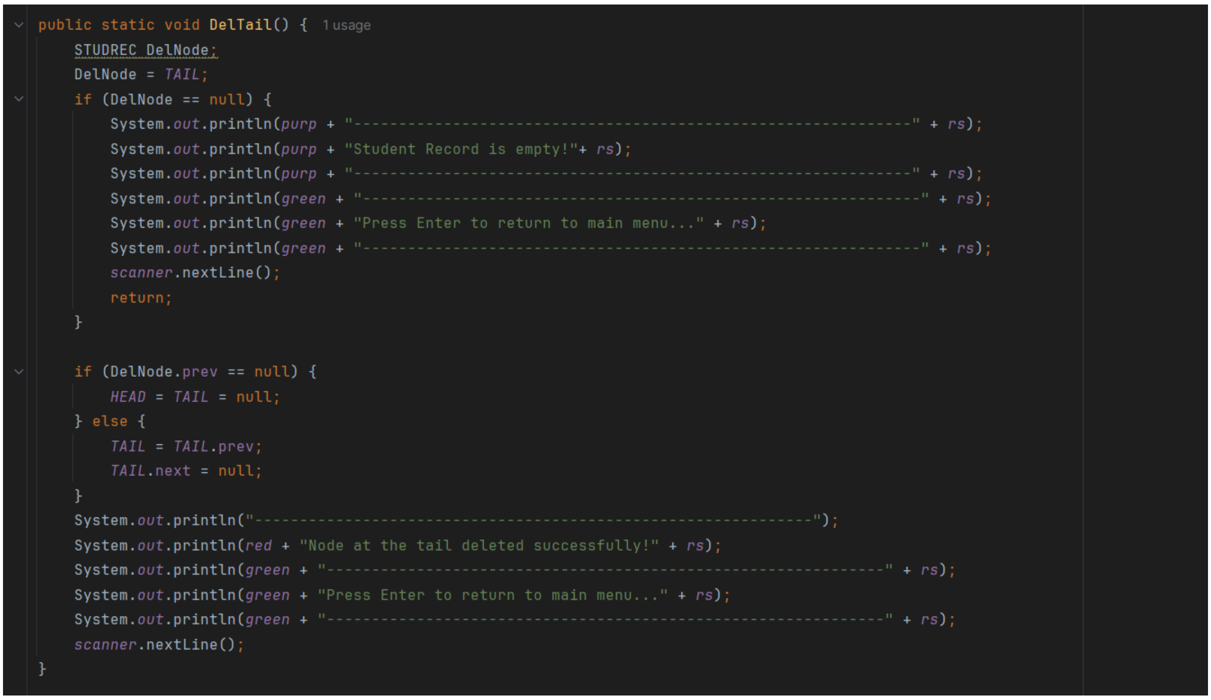
```
--------------------------------------------------------------

Enter your choice: 8

--------------------------------------------------------------


--------------------------------------------------------------
Node at the head deleted successfully!
--------------------------------------------------------------

--------------------------------------------------------------
Press Enter to return to main menu...
--------------------------------------------------------------
```



```
No.       Student No.           Student Name          Course & Year           GWA
1         1000                  Suguru Geto           Computer Science - 3    1.00

2         1001                  Gojo Satoru           Computer Science - 1    1.00

3         1002                  Toji Fushiguro        Computer Science - 2    3.00

4         1003                  Kento Nanami          Computer Science - 4    1.00

5         1004                  Toge Inumaki          Nursing - 1             1.00
```

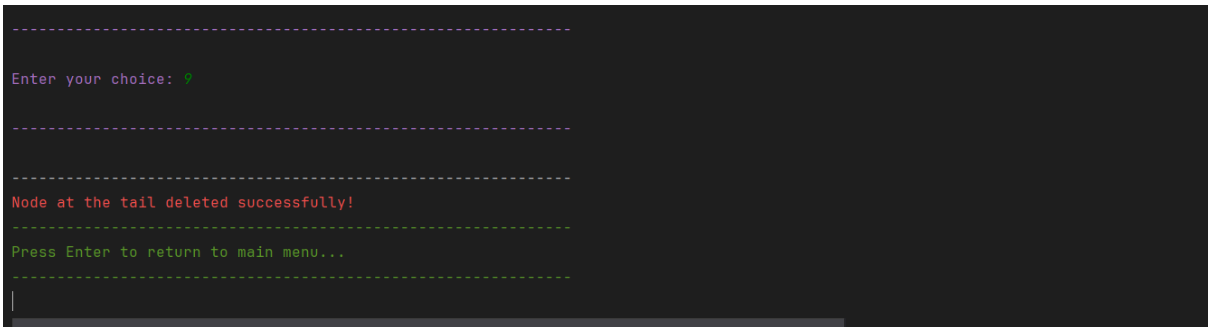
X. Deletion of Node in Double/Doubly Linked List (at the Tail)

Source Code in Java

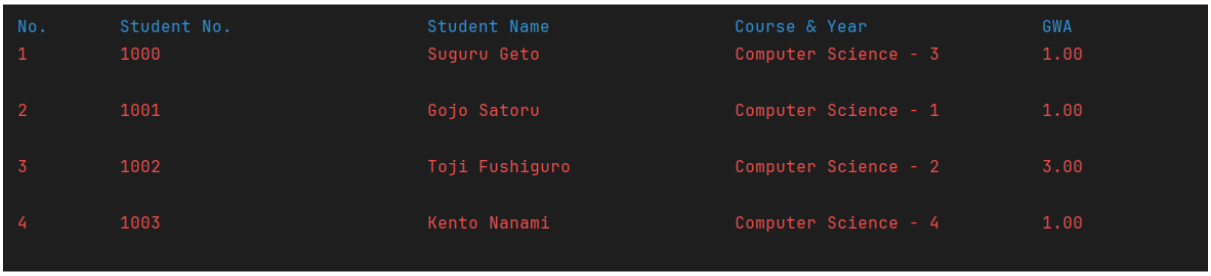


```java
public static void DelTail() {  1 usage
    STUDREC DelNode;
    DelNode = TAIL;
    if (DelNode == null) {
        System.out.println(purp + "--------------------------------------------------------------" + rs);
        System.out.println(purp + "Student Record is empty!"+ rs);
        System.out.println(purp + "--------------------------------------------------------------" + rs);
        System.out.println(green + "--------------------------------------------------------------" + rs);
        System.out.println(green + "Press Enter to return to main menu..." + rs);
        System.out.println(green + "--------------------------------------------------------------" + rs);
        scanner.nextLine();
        return;
    }

    if (DelNode.prev == null) {
        HEAD = TAIL = null;
    } else {
        TAIL = TAIL.prev;
        TAIL.next = null;
    }
    System.out.println("--------------------------------------------------------------");
    System.out.println(red + "Node at the tail deleted successfully!" + rs);
    System.out.println(green + "--------------------------------------------------------------" + rs);
    System.out.println(green + "Press Enter to return to main menu..." + rs);
    System.out.println(green + "--------------------------------------------------------------" + rs);
    scanner.nextLine();
}
```

Output



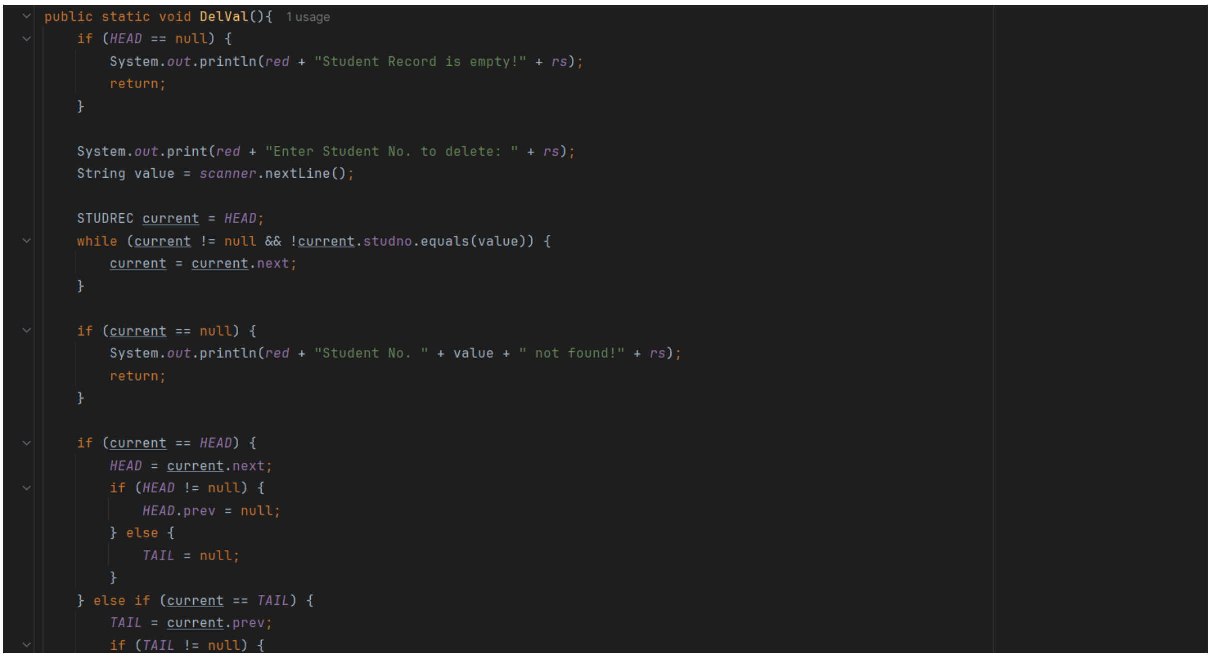
```
--------------------------------------------------------------

Enter your choice: 9

--------------------------------------------------------------

--------------------------------------------------------------
Node at the tail deleted successfully!
--------------------------------------------------------------
Press Enter to return to main menu...
--------------------------------------------------------------
|
```



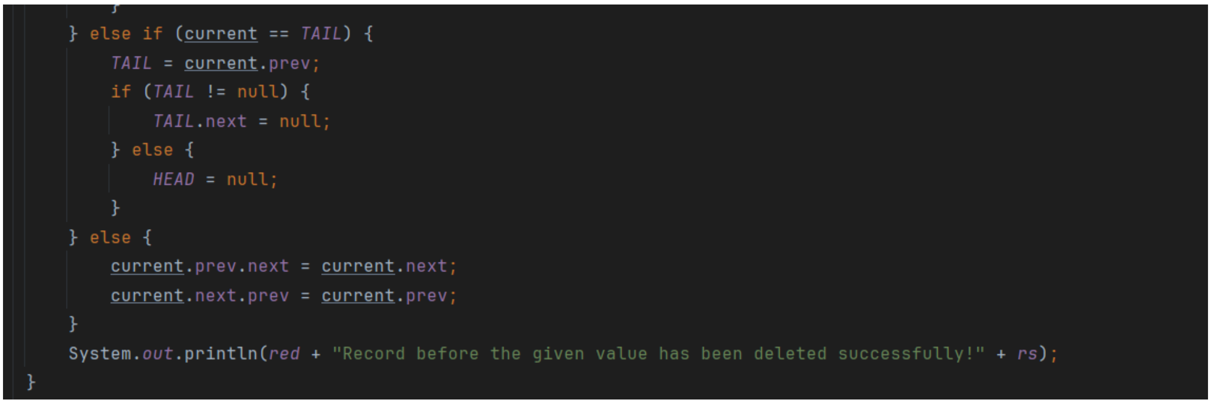
```
No.         Student No.                       Student Name                      Course & Year                     GWA
1           1000                              Suguru Geto                       Computer Science - 3              1.00

2           1001                              Gojo Satoru                       Computer Science - 1              1.00

3           1002                              Toji Fushiguro                    Computer Science - 2              3.00

4           1003                              Kento Nanami                      Computer Science - 4              1.00
```

XI. Deletion of Node in Double/Doubly Linked List (by Value)

Source Code in Java

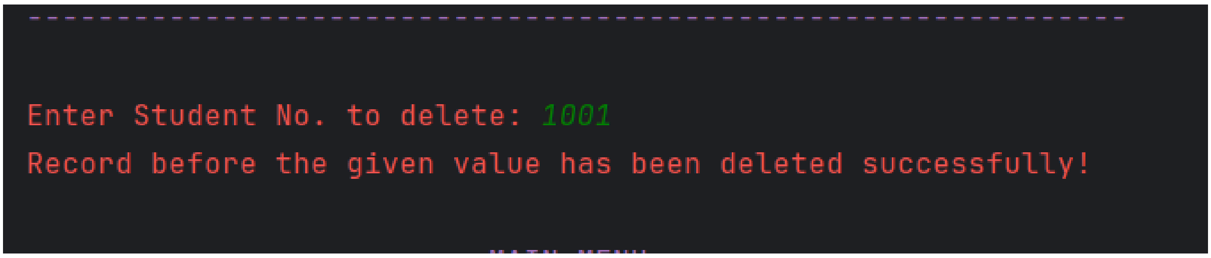


```java
public static void DelVal(){  1 usage
    if (HEAD == null) {
        System.out.println(red + "Student Record is empty!" + rs);
        return;
    }

    System.out.print(red + "Enter Student No. to delete: " + rs);
    String value = scanner.nextLine();

    STUDREC current = HEAD;
    while (current != null && !current.studno.equals(value)) {
        current = current.next;
    }

    if (current == null) {
        System.out.println(red + "Student No. " + value + " not found!" + rs);
        return;
    }

    if (current == HEAD) {
        HEAD = current.next;
        if (HEAD != null) {
            HEAD.prev = null;
        } else {
            TAIL = null;
        }
    } else if (current == TAIL) {
        TAIL = current.prev;
        if (TAIL != null) {
```

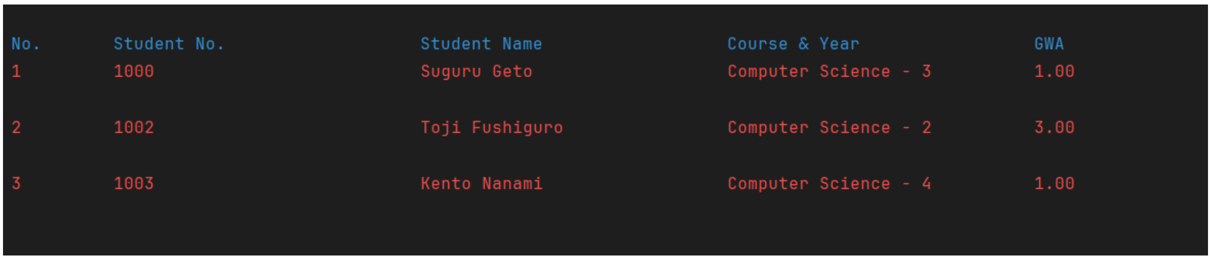


```java
        }
    } else if (current == TAIL) {
        TAIL = current.prev;
        if (TAIL != null) {
            TAIL.next = null;
        } else {
            HEAD = null;
        }
    } else {
        current.prev.next = current.next;
        current.next.prev = current.prev;
    }
    System.out.println(red + "Record before the given value has been deleted successfully!" + rs);
}
```

Output



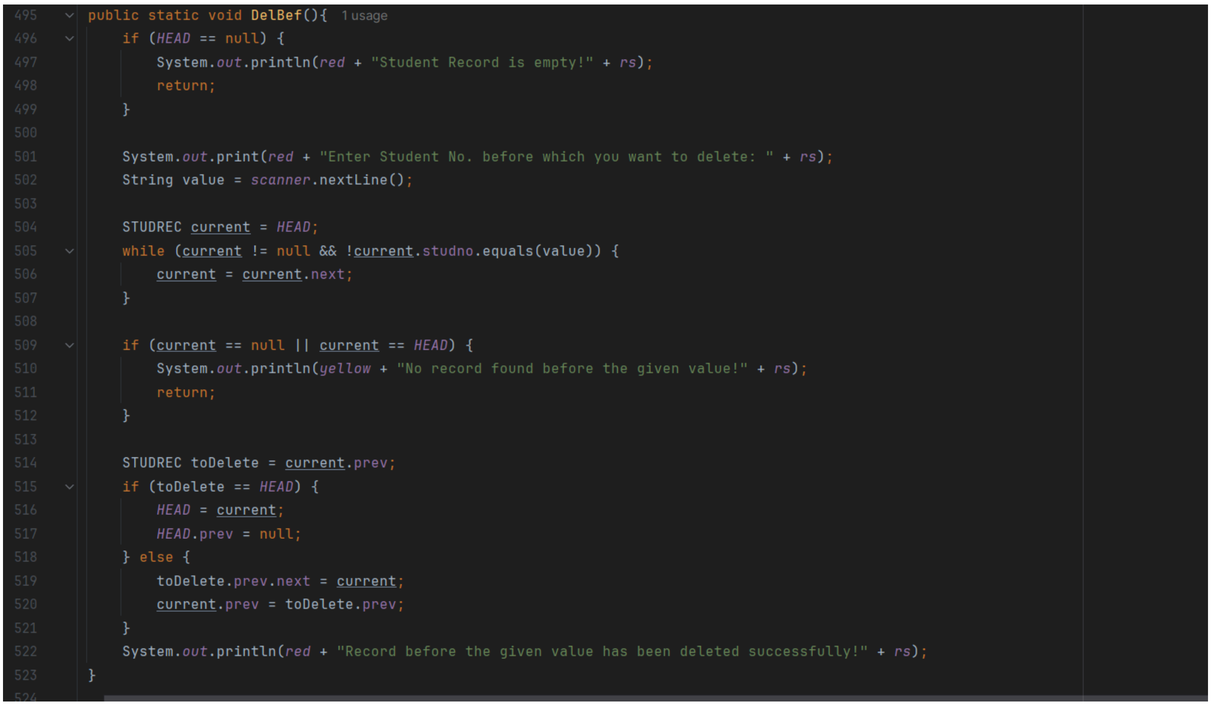
```
-----------------------------------------------


Enter Student No. to delete: 1001
Record before the given value has been deleted successfully!


                 MAIN MENU
```



```
No.     Student No.              Student Name             Course & Year            GWA
1       1000                     Suguru Geto              Computer Science - 3     1.00

2       1002                     Toji Fushiguro           Computer Science - 2     3.00

3       1003                     Kento Nanami             Computer Science - 4     1.00
```

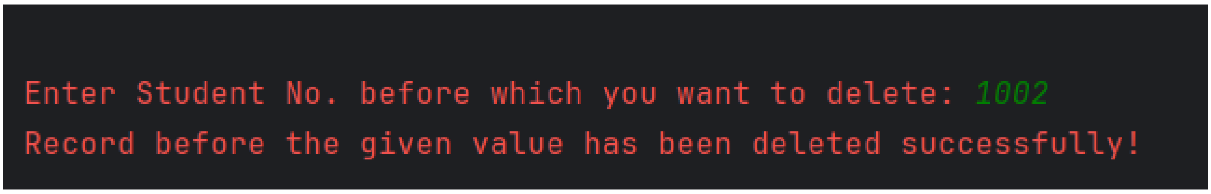
XII. Deletion of Node in Double/Doubly Linked List (Before a Node)

Source Code in Java

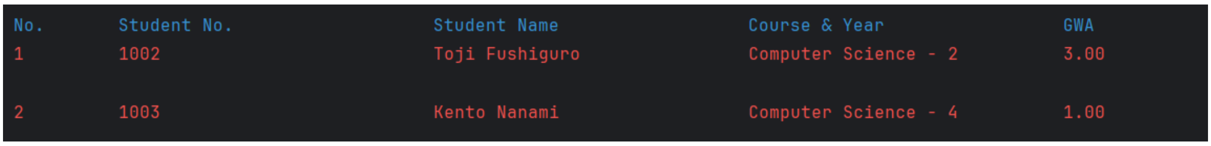


```java
public static void DelBef(){  1 usage
    if (HEAD == null) {
        System.out.println(red + "Student Record is empty!" + rs);
        return;
    }

    System.out.print(red + "Enter Student No. before which you want to delete: " + rs);
    String value = scanner.nextLine();

    STUDREC current = HEAD;
    while (current != null && !current.studno.equals(value)) {
        current = current.next;
    }

    if (current == null || current == HEAD) {
        System.out.println(yellow + "No record found before the given value!" + rs);
        return;
    }

    STUDREC toDelete = current.prev;
    if (toDelete == HEAD) {
        HEAD = current;
        HEAD.prev = null;
    } else {
        toDelete.prev.next = current;
        current.prev = toDelete.prev;
    }
    System.out.println(red + "Record before the given value has been deleted successfully!" + rs);
}
```

Output



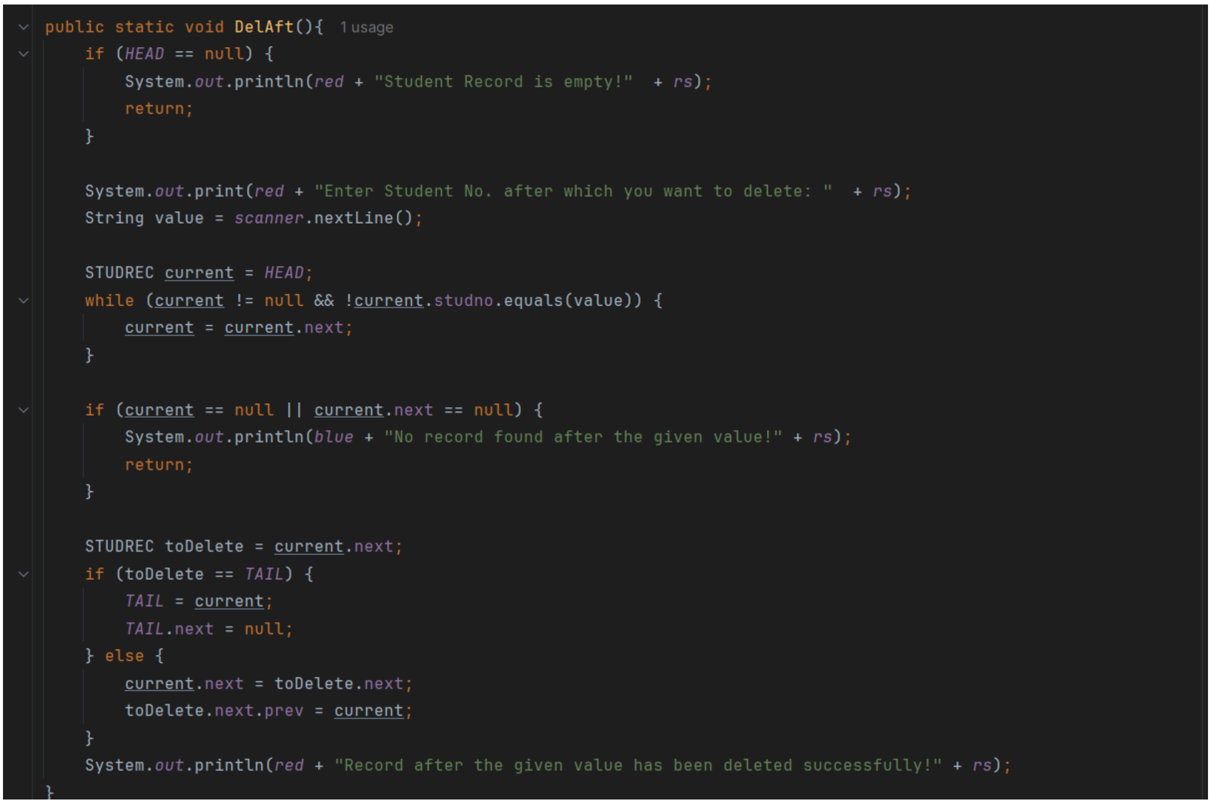
```
Enter Student No. before which you want to delete: 1002
Record before the given value has been deleted successfully!
```



```
No.       Student No.                  Student Name                 Course & Year                GWA
1         1002                         Toji Fushiguro               Computer Science - 2         3.00

2         1003                         Kento Nanami                 Computer Science - 4         1.00
```

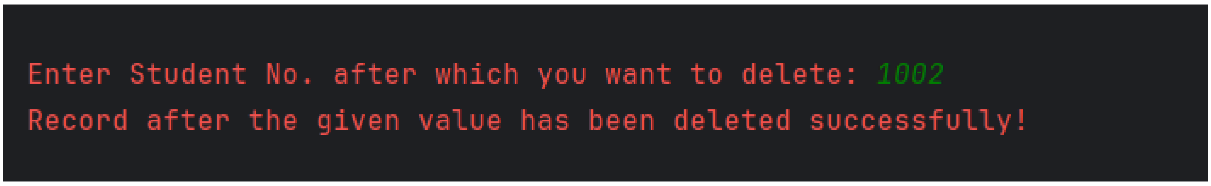
XIII. Deletion of Node in Double/Doubly Linked List (After a Node)

Source Code in Java

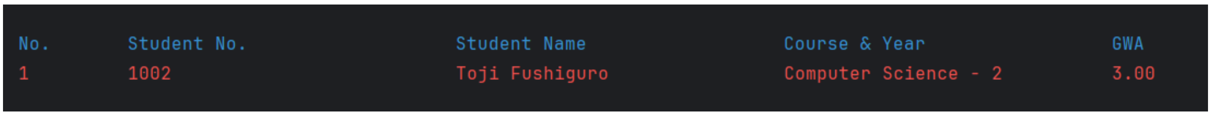


```java
public static void DelAft(){  1 usage
    if (HEAD == null) {
        System.out.println(red + "Student Record is empty!"  + rs);
        return;
    }

    System.out.print(red + "Enter Student No. after which you want to delete: "  + rs);
    String value = scanner.nextLine();

    STUDREC current = HEAD;
    while (current != null && !current.studno.equals(value)) {
        current = current.next;
    }

    if (current == null || current.next == null) {
        System.out.println(blue + "No record found after the given value!" + rs);
        return;
    }

    STUDREC toDelete = current.next;
    if (toDelete == TAIL) {
        TAIL = current;
        TAIL.next = null;
    } else {
        current.next = toDelete.next;
        toDelete.next.prev = current;
    }
    System.out.println(red + "Record after the given value has been deleted successfully!" + rs);
}
```

Output



```
Enter Student No. after which you want to delete: 1002
Record after the given value has been deleted successfully!
```



```
No.       Student No.                Student Name               Course & Year              GWA
1         1002                       Toji Fushiguro             Computer Science - 2       3.00
```