/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* file : LinkedListDemo.java

\* Author : Alwin J Thomas

\* version : 1.0

\* description : Program to implement Doubly linked list

\* date : 05/12/2023

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

PROGRAM

import java.util.Scanner;

import java.util.LinkedList;

public class LinkedListDemo {

public static void main(String[] args) {

DoublyLinkedList <Integer> dll = new DoublyLinkedList <Integer>();

Scanner scan = new Scanner(System.in);

int choice = 0;

while (choice < 6) {

System.out.print("1.Insert at Front\n2.Insert at Last\n3.Insert at any

position\n4.Remove from any position\n5.Display\n6.Exit\nEnter your choice: ");

choice = scan.nextInt();

switch (choice) {

case 1:

System.out.println("Enter the Element: ");

int numbers = scan.nextInt();

dll.insertFront(numbers);

break;

case 2:

System.out.println("Enter the Element: ");

numbers = scan.nextInt();

dll.insertLast(numbers);

break;

case 3:

System.out.println("Enter the Element: ");

numbers = scan.nextInt();

System.out.println("Enter the Position: ");

int position = scan.nextInt();

dll.insertAtAnyPosition(position,numbers );

break;

case 4:

System.out.println("Enter the Position: ");

position = scan.nextInt();

dll.removeAtAnyPos(position);

break;

case 5:

System.out.print("Doubly Linked List: ");

dll.display();

break;

default:

System.out.println("Quiting..");

break;

}

}

}

}

class DoublyLinkedList<T>{

private LinkedList<T> list = new LinkedList<>();

public void insertLast(T num) {

list.addLast(num);

}

public void insertFront (T num){

list.addFirst(num);

}

public void insertAtAnyPosition(int index, T num){

list.add(index, num);

}

public void removeAtAnyPos(int index){

list.remove(index);

}

public void display() {

for(T num: list) {

System.out.print(num +" ");

}

System.out.println();

}

}

OUTPUT

1.Insert at Front

2.Insert at Last

3.Insert at any position

4.Remove from any position

5.Display

6.Exit

Enter your choice: 1

Enter the Element: 10

1.Insert at Front

2.Insert at Last

3.Insert at any position

4.Remove from any position

5.Display

6.Exit

Enter your choice: 2

Enter the Element:

20

1.Insert at Front

2.Insert at Last

3.Insert at any position

4.Remove from any position

5.Display

6.Exit

Enter your choice: 3

Enter the Element:

15

Enter the Position:

1

1.Insert at Front

2.Insert at Last

3.Insert at any position

4.Remove from any position

5.Display

6.Exit

Enter your choice: 5

Doubly Linked List: 10 15 20

1.Insert at Front

2.Insert at Last

3.Insert at any position

4.Remove from any position

5.Display

6.Exit

Enter your choice: 4

Enter the Position: 2

1.Insert at Front

2.Insert at Last

3.Insert at any position

4.Remove from any position

5.Display

6.Exit

Enter your choice: 5

Doubly Linked List: 10 15

1.Insert at Front

2.Insert at Last

3.Insert at any position

4.Remove from any position

5.Display

6.Exit

Enter your choice: 6

Quiting..