

TD TP N° 4

Exercice N° 4 :-----

```
import java.util.*;
public class Exo4 {

    public static void main(String[] args) {
        LinkedList<Integer> l = new LinkedList<>();
        l.add(5);
        l.add(4);
        l.add(3);
        l.add(2);
        l.add(140);
        LinkedList<Integer> p = new LinkedList<>();
        p.add(0);
        p.add(3);
        p.add(12);

        System.out.println("L= " + l);
        System.out.println("P= " + p);
        printLots(l,p);

    }
    public static void printLots(LinkedList<Integer> l,LinkedList<Integer> p)
    {
        ListIterator<Integer> itl = l.listIterator();
        ListIterator<Integer> itp = p.listIterator();
        boolean b=true;
        Integer d=0;
        while(itl.hasNext() && itp.hasNext()&& b)
        {
            Integer f = itp.next();
            if (f<=l.size())
            {
                for(int i=0;i<f-d;i++)
                    itl.next();
                System.out.print(itl.next()+" ");
                itl.previous();
                d=f;
            }
            else
                b=false;
        }
    }
}
```

Exercice N° 5 :-----

```
import java.util.*;
public class Exo5 {

    public static void main(String[] args) {
        Stack<Integer> s = new Stack<>();
        s.push(103);
        s.push(2);
        s.push(98);
        s.push(11);
        s.push(7);
        System.out.println(s);
        permutationCir(s,3);
        System.out.println(s);
    }
}
```

```

    }
    public static void permutationCir(Stack<Integer> l,int n)
    {
        Stack<Integer> s = new Stack<>();
        for(int i=1;i<=n;i++)
        {
            Integer sommet = l.pop();
            while(!l.empty())
                s.push(l.pop());
            l.push(sommet);
            while(!s.empty())
                l.push(s.pop());
        }
    }
}

```

#### Exercice N° 6 :-----

```

import java.util.Scanner;
import java.util.Stack;

public class Exo6 {

    public static void main(String[] args) {

        Stack<Character> s = new Stack<>();
        Character monChar=' ';
        Scanner clavier = new Scanner(System.in);

        boolean e=true;

        while ((monChar !='#') && (e))
        {
            monChar =clavier.next().charAt(0);
            if (monChar=='(')
                s.push(monChar);
            if (monChar==')')
            {
                if (!s.empty())
                    s.pop();
                else
                    e=false;
            }
        }
        if (e)
            System.out.println("Expression bien parenthésée");
        else
            System.out.println("Expression mauvaise");
    }
}

```

#### Exercice N° 7 :-----

```

package exo7;
import java.util.*;
public class Exo8TD3 {

    public static void main(String[] args) {
        LinkedList<Integer> file =new LinkedList<>();

        file.offer(103);
        file.offer(2);
        file.offer(98);
        file.offer(11);
    }
}

```

```

        file.offer(7);
        file = permutCir(file,2);
        while(!file.isEmpty())
            System.out.print(file.poll()+" ");

    }
    static LinkedList<Integer> permutCir(LinkedList<Integer> f,int n)
    {
        int cir;
        int taille =f.size();
        cir=n%taille;
        for(int i=1;i<=cir;i++)
        {
            f.offer(f.poll());
        }
        return f;
    }
}

```