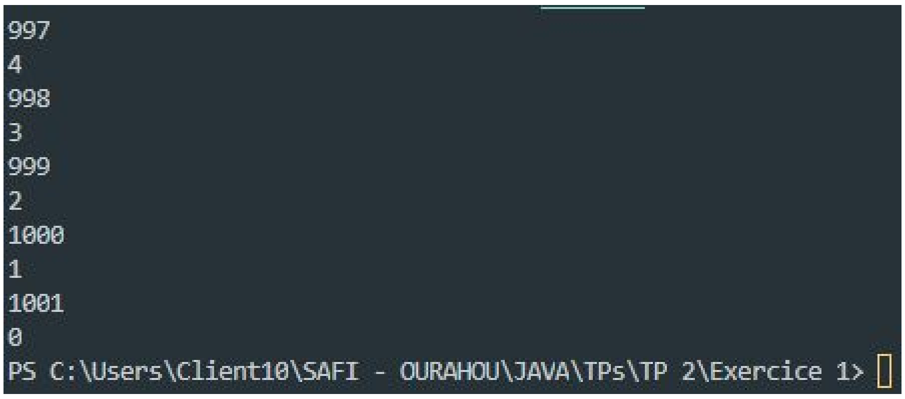
TP 2

Exercice 1:

Q1:

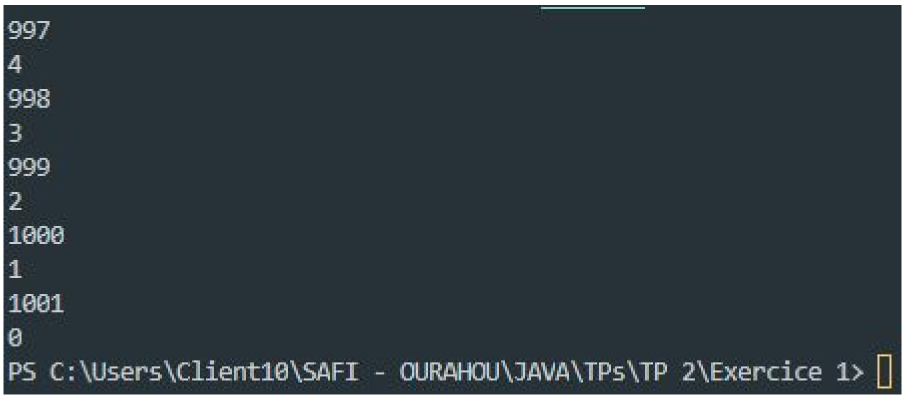
```java
public class RunnableTest implements Runnable {

    int val;

    RunnableTest(int val)
    {
        this.val = val;
    }

    @Override
    public void run() {
        // TODO Auto-generated method stub
        try{
            if(val == 1){
                while(val <= 1000) {
                    val++;
                    System.out.println(val);
                    Thread.sleep(100);
                }
            }
            else if(val == 1000){
                while(val >= 1){
                    val--;
                    System.out.println(val);
                    Thread.sleep(100);
                }
            }
            else{
                System.out.println("La valeur peut etre 1 ou 1000 !!");
            }
        }catch(InterruptedException e){
            e.printStackTrace();
            return;
        }
    }

    public static void main(String[] args) {
        Runnable rt1 = new RunnableTest(1);
        Runnable rt2 = new RunnableTest(1000);

        new Thread(rt1).start();
        new Thread(rt2).start();
    }
}
```



```
997
4
998
3
999
2
1000
1
1001
0
PS C:\Users\Client10\SAFI - OURAHOU\JAVA\TPs\TP 2\Exercice 1>
```

Q2:

```
public class ThreadsTest extends Thread {

    int val;

    ThreadsTest(int val)
    {
        this.val = val;
    }

    @Override
    public void run() {
        // TODO Auto-generated method stub
        try{
            if(val == 1){
                while(val <= 1000) {
                    val++;
                    System.out.println(val);
                    sleep(100);
                }
            }
            else if(val == 1000){
                while(val >= 1){
                    val--;
                    System.out.println(val);
                    sleep(100);
                }
            }
            else{
                System.out.println("La valeur peut etre 1 ou 1000 !!");
            }
        }catch(InterruptedException e){
            e.printStackTrace();
            return;
        }
    }

    public static void main(String[] args) {
        Thread t1 = new ThreadsTest(1);
        Thread t2 = new ThreadsTest(1000);

        t1.start();
        t2.start();
    }
}
```

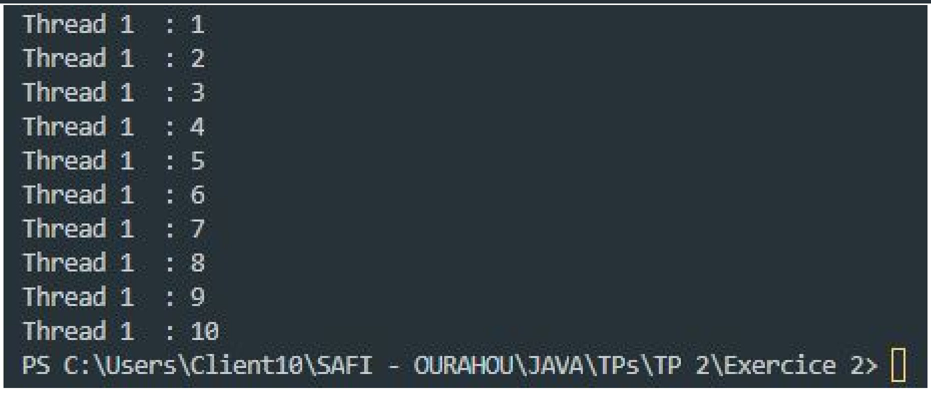


```
997
4
998
3
999
2
1000
1
1001
0
PS C:\Users\Client10\SAFI - OURAHOU\JAVA\TPs\TP 2\Exercice 1>
```

Exercice 2:

Q1:

```
public class Compteur extends Thread
{
    int maxValue;
    String nom;

    Compteur(String n,int v)
    {
        this.nom = n;
        this.maxValue = v;
    }

    @Override
    public void run() {
        // TODO Auto-generated method stub
        try{
        for(int i=1;i<=maxValue;i++)
        {
            System.out.println(this.nom+" : "+i);
            sleep(100);
        }
    }catch(InterruptedException e){
        return;
    }
    }

    public static void main(String[] args) {
        Thread t1 = new Compteur("Thread 1 ", 10);
        t1.start();
    }
}
```

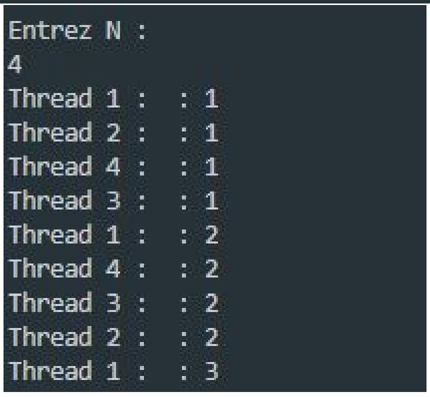


```
Thread 1  : 1
Thread 1  : 2
Thread 1  : 3
Thread 1  : 4
Thread 1  : 5
Thread 1  : 6
Thread 1  : 7
Thread 1  : 8
Thread 1  : 9
Thread 1  : 10
PS C:\Users\Client10\SAFI - OURAHOU\JAVA\TPs\TP 2\Exercice 2>
```

Q2:

```
import java.lang.*;
import java.util.Scanner;
```

```java
public class TestOrder {
    public static void main(String[] args) {

        Scanner scanner = new Scanner(System.in);
        System.out.println("Entrez N : ");

        String str = scanner.nextLine();
        int val = Integer.parseInt(str);

        for(int i=1;i<=val;i++)
        {
            new Compteur("Thread "+i+" : ", 5).start();
        }
    }
}
```



```
Entrez N :
4
Thread 1 :  : 1
Thread 2 :  : 1
Thread 4 :  : 1
Thread 3 :  : 1
Thread 1 :  : 2
Thread 4 :  : 2
Thread 3 :  : 2
Thread 2 :  : 2
Thread 1 :  : 3
```

Q3 : Oui

Q4:

```java
    public void run() {
        // TODO Auto-generated method stub
        try{
        for(int i=1;i<=maxValue;i++)
        {
            System.out.println(this.nom+" : "+i);
            sleep((1000-500)*(long)Math.random());
        }
    }catch(InterruptedException e){
        return;
    }
    }
```

Q5:

Memoire Commune;

Exercice 3:
Q1:
```java
public class Valeur{
    int valeur;

    Valeur(int vleur)
    {
        this.valeur = vleur;
    }

    public int getVal()
    {
        return this.valeur;
    }

    public void add(int i)
    {
        this.valeur += i;
    }

    public String toString(){
        return ("Valeur : "+this.getVal());
    }
}
```
Q2:
```java
import java.math.*;

public class Ajob implements Runnable
{
    Valeur myVal;
    int i;

    Ajob(Valeur val,int v)
    {
        this.myVal = val;
        this.i = v;
    }

    @Override
    public void run() {
        // TODO Auto-generated method stub
        try{
        for(int i=0;i<Math.pow(10, 6);i++)
        {
            myVal.add(this.i);
            System.out.println(myVal.toString());
            Thread.sleep(5);
        }
        }catch(InterruptedException e){return;}
    }
}
```

Q3:

```java
public class Test {
    public static void main(String[] args) {
        Valeur myval = new Valeur(5);

        Ajob a1 = new Ajob(myval,1);
        Ajob a2 = new Ajob(myval,-1);

        new Thread(a1).start();
        new Thread(a2).start();
    }
}
```

Q4 : la meme valeur stockee au premiere fois.

Q5 : la concurrence se trouve entre les threads,le premier incremente la valeur et l’autre la decremente.

Q8:

```java
public class Valeur{
    int valeur;

    Valeur(int vleur)
    {
        this.valeur = vleur;
    }

    public int getVal()
    {
        return this.valeur;
    }

    public synchronized void add(int i)
    {
        this.valeur += i;
    }

    public String toString(){
        return ("Valeur : "+this.getVal());
    }
}
```