TP 2 - Solution

Exercice 1 :

Partie 1 :

```
import java.net.* ;
import java.io.* ;
public class Process1 {
   public static void main (String args[]) {
      try {
         for(int i=1; i<=10; i++) {
            DatagramSocket ds = new DatagramSocket(7000);
            InetAddress AdrP1 = InetAddress.getByName("127.0.0.1");
            String s = "message "+i+" de P1";
            byte[] m = s.getBytes();
            DatagramPacket dp = new DatagramPacket(m, m.length, AdrP1, 8000);
            ds.send(dp) ;
            byte[] tampon = new byte[1000];
            DatagramPacket dp1 = new DatagramPacket(tampon, tampon.length);
            ds.receive(dp1);
            String s1= new String(dp1.getData());
            System.out.println(s);
            ds.close();
         }
      }
      catch (UnknownHostException e) { }
      catch (SocketException ee) { }
      catch (IOException eee) { }
   }
}
```

```
import java.net.* ;
import java.io.* ;
public class Process2 {
   public static void main (String args[]) {
      try {
         for(int j=1; j<=10; j++) {
            DatagramSocket ds= new DatagramSocket(8000);
            InetAddress AdrP2 = InetAddress.getByName("127.0.0.1");
            String s = "message "+j+" de P2";
            byte[] m = s.getBytes();
```

```
            DatagramPacket dp1 = new DatagramPacket(m, m.length, AdrP2, 7000);
            ds.send(dp1);
            byte[] tampon = new byte[1000];
            DatagramPacket dp = new DatagramPacket(tampon, tampon.length);
            ds.receive(dp);
            String s1 = new String(dp.getData());
            System.out.println(s1);
            ds.close();
         }
      }
      catch (UnknownHostException e) { }
      catch (SocketException ee) { }
      catch (IOException eee) { }
   }
}
```

Partie 2 :

```
import java.net.* ;
import java.io.* ;
public class Process1 {
   public static void main (String args[]) {
      try {
         int T=10;
         for(int i=1; i<=10; i++) {
            DatagramSocket ds = new DatagramSocket(7000);
            Thread.sleep (T*1000);
            InetAddress AdrP1 = InetAddress.getByName("127.0.0.1");
            String s = "message "+i+" de P1";
            byte[] m = s.getBytes();
            DatagramPacket dp = new DatagramPacket(m, m.length, AdrP1, 8000);
            ds.send(dp) ;
            byte[] tampon = new byte[1000];
            DatagramPacket dp1 = new DatagramPacket(tampon, tampon.length);
            ds.receive(dp1);
            String s1 = new String(dp1.getData());
            System.out.println(s1);
            T=10;
            ds.close();
         }
      }
      catch (UnknownHostException e) { }
      catch (SocketException ee) { }
      catch (IOException eee) { }
      catch (InterruptedException eeee) { }
   }
}
```

```
import java.net.* ;
import java.io.* ;
public class Process2 {
   public static void main (String args[]) {
      try {
         int T=5;
         for(int j=1; j<=10; j++) {
            DatagramSocket ds= new DatagramSocket(8000);
            Thread.sleep (T*1000);
            InetAddress AdrP2 = InetAddress.getByName("127.0.0.1");
            String s = "message "+j+" de P2";
            byte[] m = s.getBytes();
            DatagramPacket dp = new DatagramPacket(m, m.length, AdrP2, 7000);
            ds.send(dp);
            byte[] tampon = new byte[1000];
            DatagramPacket dp1 = new DatagramPacket(tampon, tampon.length);
            ds.receive(dp1);
            String s1 = new String(dp1.getData());
            System.out.println(s1);
            T=5;
            ds.close();
         }
      }
      catch (UnknownHostException e) { }
      catch (SocketException ee) { }
      catch (IOException eee) { }
      catch (InterruptedException eeee) { }
   }
}
```

Exercice 2 :

```
import java.net.*;
import java.io.*;
import java.nio.ByteBuffer;
import java.util.Scanner;
public class Process1 {
   public static void main (String args[]) {
      try {
         DatagramSocket ds = new DatagramSocket(7000);
         InetAddress AdrP1 = InetAddress.getByName("127.0.0.1");
         Scanner clavier = new Scanner(System.in);
         int a = clavier.nextInt();
         int b = clavier.nextInt();
```

```
        int c = clavier.nextInt();
        if (a==0) {
           byte[] x = ByteBuffer.allocate(4).putInt(b).array();
           DatagramPacket dp1 = new DatagramPacket(x, x.length, AdrP1, 8000);
           ds.send(dp1);
           byte[] y = ByteBuffer.allocate(4).putInt(c).array();
           DatagramPacket dp2 = new DatagramPacket(y, y.length, AdrP1, 8000);
           ds.send(dp2);
        }
        else {
           byte[] x = ByteBuffer.allocate(4).putInt(a).array();
           DatagramPacket dp1 = new DatagramPacket(x, x.length, AdrP1, 9000);
           ds.send(dp1);
           byte[] y = ByteBuffer.allocate(4).putInt(b).array();
           DatagramPacket dp2 = new DatagramPacket(y, y.length, AdrP1, 9000);
           ds.send(dp2);
           byte[] z = ByteBuffer.allocate(4).putInt(c).array();
           DatagramPacket dp3 = new DatagramPacket(z, z.length, AdrP1, 9000);
           ds.send(dp3);
           byte[] tampon = new byte[1000];
           DatagramPacket dp4 = new DatagramPacket(tampon, tampon.length);
           ds.receive(dp4);
           int Delta = ByteBuffer.wrap(dp4.getData()).getInt();
           System.out.println("Delta = "+Delta);
           if (Delta<0)
              System.out.println("Pas de solution");
           else {
              double x1=(-b-Math.sqrt(Delta)/2*a);
              System.out.println("x1 = "+x1);
              double x2=(-b+Math.sqrt(Delta)/2*a);
              System.out.println("x2 = "+x2);
           }
        }
        ds.close();
     }
     catch (UnknownHostException e) { }
      catch (SocketException ee) { }
      catch (IOException eee) { }
  }
}
```

```
import java.net.*;
import java.io.*;
import java.nio.ByteBuffer;
public class Process2 {
   public static void main (String args[]) {
      try {
            DatagramSocket ds= new DatagramSocket(8000);
            byte[] tampon1 = new byte[1000];
            DatagramPacket dp1 = new DatagramPacket(tampon1, tampon1.length);
            ds.receive(dp1);
            int a = ByteBuffer.wrap(dp1.getData()).getInt();
            byte[] tampon2 = new byte[1000];
            DatagramPacket dp2 = new DatagramPacket(tampon2, tampon2.length);
            ds.receive(dp2);
            int b = ByteBuffer.wrap(dp2.getData()).getInt();
            if (a==0)
               if (b==0)
                  System.out.println("La solution est R");
               else
                  System.out.println("Pas de solution");
            else{
               double x = -b/a;
               System.out.println("x = "+x);
               ds.close();
            }
         }
      catch (SocketException e) { }
      catch (IOException ee) { }
   }
}

import java.net.*;
import java.io.*;
import java.nio.ByteBuffer;
public class Process3 {
   public static void main (String args[]) {
      try {
            DatagramSocket ds= new DatagramSocket(9000);
            byte[] tampon1 = new byte[1000];
            DatagramPacket dp1 = new DatagramPacket(tampon1, tampon1.length);
            ds.receive(dp1);
            int a = ByteBuffer.wrap(dp1.getData()).getInt();
            byte[] tampon2 = new byte[1000];
```

```
            DatagramPacket dp2 = new DatagramPacket(tampon2, tampon2.length);
            ds.receive(dp2);
            int b = ByteBuffer.wrap(dp2.getData()).getInt();
            byte[] tampon3 = new byte[1000];
            DatagramPacket dp3 = new DatagramPacket(tampon3, tampon3.length);
            ds.receive(dp3);
            int c = ByteBuffer.wrap(dp3.getData()).getInt();
            int Delta = b*b-4*a*c;
            byte[] d = ByteBuffer.allocate(4).putInt(Delta).array();
            InetAddress AdrP3 = InetAddress.getByName("127.0.0.1");
            DatagramPacket dp4 = new DatagramPacket(d, d.length, AdrP3, 7000);
            ds.send(dp4);
            ds.close();
          }
        catch (SocketException e) { }
        catch (IOException ee) { }
    }
}
```