



Haute école d'ingénierie et d'architecture Fribourg
Hochschule für Technik und Architektur Freiburg

PROGRAMMATION ET TP

T1A

RÉSEAU ET SÉCURITÉ

S21 Stream

ROTEN MARC

2017/2018

Table des matières

1	Exercice 1	2
2	Ex2	2
3	Ex3	3
4	Ex4	3
5	Ex5	4
5.1	a	4
5.2	b	4
5.3	c	4
6	ex6	5

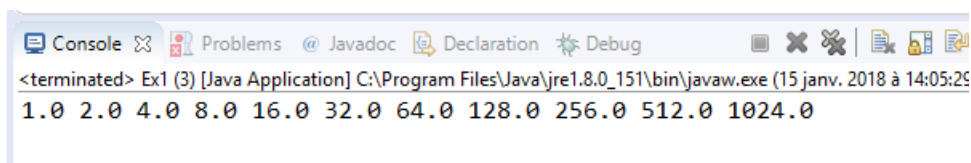
1 Exercice 1

```
package s21;

import java.util.Arrays;
import java.util.List;

public class Ex1 {
    public static void main(String[] args) {
        List<Integer> listEx1 =
            Arrays.asList(0,1,2,3,4,5,6,7,8,9,10);
        listEx1.stream().forEach(s-> {
            System.out.print(Math.pow(2,s)+" ");
        });
    }
}
```

retour console



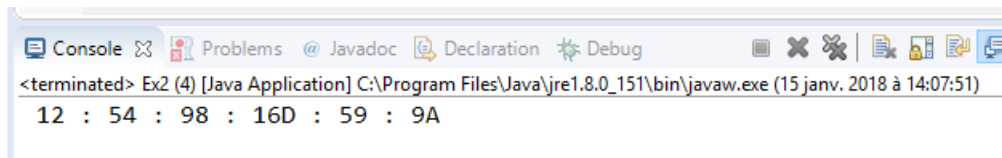
2 Ex2

```
package s21;

import java.util.Arrays;
import java.util.List;

public class Ex2 {
    public static void main(String[] args) {
        List<Integer> myList =
            Arrays.asList(18,84,152,365,89,154);
        myList.stream().forEach((e)->{
            if(e==myList.get(myList.size()-1)) {
                System.out.print(String.format(" %02X ", e));
            }else {
                System.out.print(String.format(" %02X :", e));
            }
        });
    }
}
```

retour console



3 Ex3

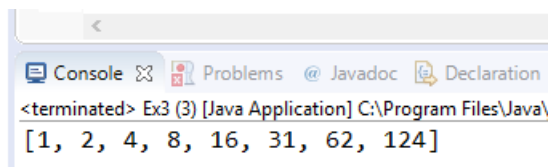
```
package s21;

import java.util.Arrays;
import java.util.stream.LongStream;

public class Ex3 {

    public static void main(String[] args) {
        System.out.println(Arrays.toString(divisors(496L)));
    }
    public static long[] divisors(long x) {
        return LongStream.range(1, x/2).filter(e ->
            x%e==0).toArray();
    }
}
```

retour console



4 Ex4

```
package s21;

import java.util.Arrays;
import java.util.stream.LongStream;

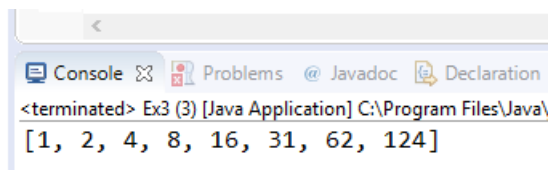
public class Ex4 {
    public static void main(String[] args) {
        System.out.println(sum(Ex3.divisors(496L)));
    }
}
```

```

    }
    private static long[] perfectNumber(long l) {
        return LongStream .range(1, l)
            .filter(x -> x==sum(Ex3.divisors(x)))
            .toArray();
    }
    private static char[] sum(long[] divisors) {
        return Arrays .stream(1)
            .sum();
    }
}

```

retour console



5 Ex5

5.1 a

La methode ci-contre sert à filtrer, ligne après ligne les mots contenus dans le document wordlist.txt. On commence par prendre tous les mot de longueur 11, ensuite ceux dont la deuxième lèttre est t, ensuite ceux dont la 4ème lettre est l. On traite ensuite les doublons, il faut qu'il y aie 6 doublons de notre mot. ensuite on le print

5.2 b

non, l'ordre des filtres ne change pas le résultat final, la performance pourrait êtrer affectée, il faut aller du plus spécifique au moins spécifique.

5.3 c

```

public class Exo5 {
    public static void main(String[] args){

        Path path = FileSystems.getDefault().getPath("wordlist.txt");
        try {
            Files.lines(path)
                .filter(i -> i.equals(new
                    StringBuilder(i).reverse().toString()))

```

```

        .forEach(System.out::println);
    } catch (IOException e) {
        e.printStackTrace();
    }
}
}

```

6 ex6

```

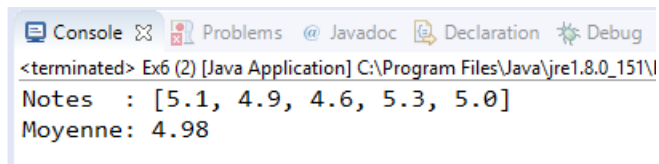
package s21;

import java.util.Arrays;

public class Ex6 {
    public static void main(String[] args) {
        double resultats[][] = {
            { 9, 10, 8, 5, 9},
            { 5, 9, 9, 8, 8},
            { 4, 8, 10, 9, 5},
            { 8, 10, 8, 10, 7},
            { 8, 9, 7, 10, 6},
        };
        double[] StreamofMyArray = Arrays.stream(resultats)
            .mapToDouble(s->Arrays.stream(s).sum())
            .map(s->1+s/10)
            .toArray();
        double moy =
            Arrays.stream(StreamofMyArray).average().getAsDouble();
        System.out.printf("Notes : %s\n",
            Arrays.toString(StreamofMyArray));
        System.out.printf("Moyenne: %.2f\n", moy);
    }
}

```

retour console



```

<terminated> Ex6 (2) [Java Application] C:\Program Files\Java\jre1.8.0_151\
Notes   : [5.1, 4.9, 4.6, 5.3, 5.0]
Moyenne: 4.98

```