Aucun document. Aucune machine. Le barème est indicatif. Les sept exercices sont indépendants.
Une question peut toujours être traitée en utilisant les précédentes (traitées ou non).
Les morceaux de code Java devront être clairement présentés, indentés et commentés.
Les classes `String`, `Scanner` et `System` sont interdites (sauf exercice 2 et question 4 de l'exercice 5).

Exercice 1. (1 point) Relever les erreurs susceptibles d'être détectées lors de la compilation de ces programmes :

```java
class Diviseur{
    static int diviseur(int n){
        int n;
        for(i=2;i<=n;i++)
            if(n%i==0)
                return i;
    }
}
```

```java
class Tableau{
    static float[] copie(double[] t){
        double[] s = new double[s.length];
        for(int i=1; s.length; i++)
            s[i] = t[i];
        return s;
    }
}
```

Exercice 2. (2 points) Les deux questions suivantes sont indépendantes.

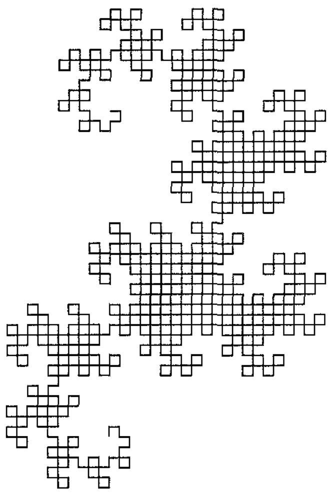
1. Écrire une fonction `stringAdmis` qui prend en arguments un numéro d'exercice et un numéro de question de cet examen et teste s'il est possible d'y utiliser la classe `String`.
2. Écrire une fonction `string` qui prend en arguments une chaîne de caractères `w` et un entier `k` et renvoie une nouvelle chaîne de caractères de longueur `k` correspondant à la troncature ou à la complétion de `w` comme dans les exemples suivants :

```
string("abc",-5) renvoie la chaîne "␣␣abc"      string("abc",0) renvoie la chaîne ""
string("abc",-4) renvoie la chaîne "␣abc"       string("abc",1) renvoie la chaîne "a"
string("abc",-3) renvoie la chaîne "abc"        string("abc",2) renvoie la chaîne "ab"
string("abc",-2) renvoie la chaîne "bc"         string("abc",3) renvoie la chaîne "abc"
string("abc",-1) renvoie la chaîne "c"          string("abc",4) renvoie la chaîne "abc␣"
```

Exercice 3. (2,5 points) Au musée de Trevors, les tarifs d'entrée sont définis par les règles suivantes :

- pour les billets individuels : gratuité pour les enfants d'au plus 3 ans, tarif réduit (3€) pour les enfants entre 4 et 10 ans et les personnes âgées à partir de 65 ans et tarif plein (5€) pour les autres.
- pour les billets de groupes (au moins 8 personnes) : sans considération d'âge, 10% de réduction sur le tarif plein

Exercice 5. (4 points) La *courbe du dragon* est une courbe récursive dont le nom provient de sa ressemblance avec une créature mythique. Elle peut être construite en représentant un virage à gauche par `true` et un virage à droite par `false`. Chaque courbe d'ordre donné peut ainsi être codée sous forme d'un tableau de booléens. La courbe du premier ordre est codée par | `false` |. Pour la courbe d'ordre `n>1`, on concatène le codage de la courbe `c` d'ordre `n-1`, celui de la courbe d'ordre `1` et celui de cette même courbe `c` mais parcourue en sens inverse (en particulier, les virages gauche et droite y sont échangés). Par exemple, la courbe du second ordre est codée par | `false` | `false` | `true` |, et celle du troisième ordre par | `false` | `false` | `true` | `false` | `false` | `true` | `true` |.



1. Donner le tableau de booléens codant la courbe du quatrième ordre. Dessiner celle-ci.
2. Écrire une fonction `suivant` qui prend une courbe en paramètre et renvoie la courbe d'ordre immédiatement supérieur.
3. Écrire une fonction récursive `dragon` qui renvoie la courbe de l'ordre en paramètre.
4. Le dessin ci-contre est réalisé via une instruction de la forme `(0,0)-(1,0)-(1,-1)-(0,-1)-(0,-2)-···` (les deux premières coordonnées sont fixées). Écrire une fonction `afficherInstructionDragon` qui affiche l'instruction associée à la courbe dont l'ordre est donné en paramètre.

Exercice 6. (4 points) Le *point d'équilibre* d'un tableau d'entiers strictement positifs est l'entier i tel que la somme des éléments du tableau jusqu'à l'élément d'indice i inclus est égale à la somme des éléments suivants. Par exemple, le point d'équilibre du tableau `t0`= | 4 | 2 | 3 | 2 | 1 | est `1`, alors que | 1 | 2 | 2 | n'en a pas.

1. On peut utiliser deux méthodes auxiliaires qui calculent les sommes partielles.
   (a) Écrire deux méthodes `sommeGauche` et `sommeDroite` qui prennent comme argument un tableau d'entiers et renvoient le tableau des sommes partielles en partant de la gauche et de la droite, respectivement. Pour `t0`, on obtient | 4 | 6 | 9 | 11 | 12 | et | 12 | 8 | 6 | 3 | 1 | respectivement.
   (b) En déduire une méthode `pointEquilibre` qui prend comme argument un tableau d'entiers (que l'on supposera tous positifs), utilise les méthodes `sommeGauche` et `sommeDroite` pour calculer et renvoyer son point d'équilibre s'il existe et `-1` sinon.
   (c) Lister les sommes, affectations et comparaisons produites par l'appel de `pointEquilibre` sur `t0`.
2. Proposer une méthode `pointEquilibreM` minimisant le nombre de sommes, affectations et comparaisons.

Exercice 7. (4 points) Pour tout entier positif `n`, on se propose de construire un carré magique de côté de longueur `c=2n+1` selon la *méthode du losange* due à Conway (voir la construction pour `n=3` ci-dessous) :

- les entiers **impairs** entre `1` et c^2 sont positionnés le long de diagonales d'un losange inscrit dans le carré ;
- les entiers **pairs** entre `1` et c^2 sont positionnés en poursuivant les diagonales ;
- le tout est glissé dans le carré en identifiant les côtés gauche et droit et les côtés haut et bas.

	0	1	2	3	4	5	6
0				43			
1			29	37	45		
2		15	23	31	39	47	
3	1	9	17	25	33	41	49
4		3	11	19	27	35	
5			5	13	21		
6				7			

`losangeImp(3)`

	0	1	2	3	4	5	6			
0				43						
1			29	37	45					
2		15	23	31	39	47				
3	1	9	17	25	33	41	49			
4		3	11	19	27	35	36	44		
5			5	13	21	22	30	38	46	
6				7	8	16	24	32	40	48
					2	10	18	26	34	42
						4	12	20	28	
							6	14		

	0	1	2	3	4	5	6
0	26	34	42	43	2	10	18
1	20	28	29	37	45	4	12
2	14	15	23	31	39	47	6
3	1	9	17	25	33	41	49
4	44	3	11	19	27	35	36
5	38	46	5	13	21	22	30
6	32	40	48	7	8	16	24

`losange(3)`

1. Construire à la main les carrés magiques selon la méthode de Conway pour `n=1` et pour `n=2`.
2. Écrire une fonction `losangeImp` qui prend un entier positif `n` en argument et retourne un carré de côté `2n+1` (sous forme de tableau bidimensionnel) dont le losange central est rempli selon la méthode de Conway.
3. Préciser comment modifier `losangeImp` en une fonction `losange` qui complète la construction de Conway.