

Aucun document. Aucune machine. Le barème est indicatif.

**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.**

Exercice 1. (2 points) Expliquer ce que produit l'exécution de chacun des deux programmes suivants :

```

1 class Omfang{
2     public static void main(String[] args){
3         int a = 3, b = 6+a;
4         System.out.println(a+" "+b);
5         if(a/b<b/a){
6             int c;
7             c = b+a; b = b+c; a = b-c;
8             System.out.println(a+" "+b+" "+c);
9         }else{
10            int c;
11            c = b-a; b = b-c; a = b+c;
12            System.out.println(a+" "+b+" "+c);
13        }
14        int c;
15        c = b+a; b = b+c; a = b-c;
16        System.out.println(a+" "+b+" "+c);
17    }

```

```

2 class Aufruf{
3     static int f(int k){
4         System.out.println("f_prend_"+k);
5         return k+g(k/3);
6     }
7     static int g(int k){
8         System.out.println("g_prend_"+k);
9         k = k*2;
10        return k-1;
11    }
12    public static void main(String[] args){
13        int k = 4;
14        k = f(k)+1;
15        System.out.println("resultat_"+k);
16        k = g(f(g(7)));
17        System.out.println("resultat_"+k);
18    }

```

Exercice 2. (2 points) Proposer des programmes dont la compilation provoque les messages d'erreurs suivants :

<pre> Skat.java:5: missing return statement } ^ Skat.java:8: cannot find symbol symbol : variable prout location: class java.lang.System System.prout.println(u); ^ 2 errors </pre>	<pre> Klud.java:5: incompatible types found : boolean required: int u=z; ^ Klud.java:6: variable y might not have been initialized System.out.println(y/0); ^ 2 errors </pre>
--	---

Exercice 3. (2,5 points) Pour déterminer le jour de la semaine correspondant à une date (jour, mois, année), on calcule les entiers s et d correspondant aux deux chiffres de gauche et aux deux chiffres de droite de l'entier année. On calcule ensuite le reste r de la division de $\text{mois} + 9$ par 12 et l'entier k défini par la formule

$$\text{jour} + 366 + \left\lfloor \frac{2 + 13r}{5} \right\rfloor + \left\lfloor \frac{s}{4} \right\rfloor - 2s + \left\lfloor \frac{d}{4} \right\rfloor + d$$

dans laquelle $\left\lfloor \frac{p}{q} \right\rfloor$ désigne le quotient de la division de p par q . Pour janvier ou février d'une année non bissextile, on soustrait 1 à k . Pour janvier ou février d'une année bissextile, on soustrait 2 à k . Le jour de la semaine (de 0 pour dimanche à 6 pour samedi) est alors le reste de la division de k par 7.

Écrire une fonction `numeroJour` qui prend en argument une date donnée par des entiers `jour`, `mois`, `annee` et renvoie le numéro du jour de la semaine. On supposera que la date est valide et que la classe `Bissextile` contient la définition d'une méthode `estBissextile`.

Exercice 4. (1,5 points) Préciser quelle erreur empêche la compilation de la classe ci-dessous. Expliquer ce que produirait son exécution après mise en commentaire de la ligne erronée, sauvegarde et compilation.

```

class Soixante17{
2     public static void main(String[] args){
3         int sept=7, dix=10, soixante=60;
4         System.out.println("_soixante-dix-sept_+_soixante-sept_");
5         System.out.println("_soixante-dix-sept_" + soixante-sept );
6         System.out.println( soixante-dix-sept + soixante-sept );
7         System.out.println( soixante-dix-sept + "_soixante-sept_" );
8     }
}

```

Exercice 5. (1 point) Préciser comment la classe ci-dessous permet de calculer la partie entière de -3.14 :

```

class PartieEntiereRecursive {
2   static int partieEntiere(double x){
        if(x>=0 && x<1) return 0;
4       if(x>0) return partieEntiere(x-1)+1;
        return partieEntiere(x+1)-1;
6   }
}

```

Cette classe pourra être utilisée dans les exercices 6 et 7.

Exercice 6. (5 points) Pour faire des beignets, il faut 400g de farine, deux oeufs, 33 cl de lait, de l'huile et de la levure. On suppose ici que l'on a toujours suffisamment d'huile et de levure.

1. Sans utiliser la classe `Math`, écrire une fonction `minimum` qui prend en arguments trois nombres réels et renvoie le minimum des trois.
2. Sans utiliser la classe `Math`, écrire une fonction `recette` qui prend en arguments une quantité de farine (en g), un nombre d'oeufs, un volume de lait (en cl) et renvoie sous forme de chaîne de caractères la recette de la pâte à beignets dans les proportions optimales de sorte que l'on puisse préparer le plus de beignets possible en fonction des ingrédients disponibles. Naturellement, le nombre d'oeufs doit être entier.
3. Écrire une fonction `main` qui demande les quantités disponibles de farine, d'oeufs et de lait, puis affiche la recette optimale du beignet.

```

$ java Beignet
Farine disponible (en g) : 500
Oeufs disponibles : 1
Lait disponible (en cl) : 120
Recette optimale du beignet:
    200.0g de farine
    1 oeuf(s)
    16.5cl de lait
    huile
    levure

```

```

$ java Beignet
Farine disponible (en g) : 1100
Oeufs disponibles : 7
Lait disponible (en cl) : 100
Recette optimale du beignet:
    1000.0g de farine
    5 oeuf(s)
    82.5cl de lait
    huile
    levure

```

4. Cette recette pour beignets peut être allégée en remplaçant en partie (ou en totalité) le lait par de la bière. Toujours dans la recherche d'une recette optimale, proposer une modification des fonctions `recette` et `main` précédentes. Donner alors un exemple d'exécution.

Exercice 7. (8 points) On considère un jeu de dés se jouant sur trois tours.

À chaque tour, le joueur

- jette trois dés à six faces (numérotées de 1 à 6),
 - calcule la somme s des numéros obtenus,
 - décide d'éliminer un ou plusieurs chiffres (entre 1 et 9) de somme s .
- Le but est de minimiser la somme des chiffres non éliminés.

1. Écrire une fonction `jetDeDe` qui renvoie un entier tiré au hasard entre 1 et 6. On pourra utiliser la fonction `random` de la classe `Math` qui renvoie un réel aléatoire supérieur à 0 et strictement inférieur à 1. En déduire une fonction `jet3Des` qui détermine trois entiers tirés au hasard entre 1 et 6, les affiche et renvoie leur somme.
2. Écrire quatre fonctions **récurives** :
 - `puissance` qui prend en arguments deux entiers positifs a et b et renvoie l'entier a élevé à la puissance b .
 - `sommeChiffres` qui prend un entier en argument et renvoie la somme de ses chiffres.
 - `chiffre` qui prend en arguments deux entiers positifs a et r et renvoie le chiffre de rang r de a (rang 1 pour le chiffre des unités, rang 2 pour celui des dizaines, etc).
 - `zero` qui prend en arguments deux entiers positifs a et c et renvoie l'entier obtenu à partir de a auquel les chiffres des rangs donnés par les chiffres de c ont été remplacés par des zéros.
3. Écrire une fonction `tour` qui prend un entier a en argument, affiche cet entier entre deux lignes de dix tirets, affiche le résultat du jet de trois dés, demande quels chiffres éliminer sous la forme d'un entier c et, si la somme des chiffres de c correspond bien au jet des trois dés, renvoie l'entier obtenu à partir de a auquel les chiffres des rangs donnés par les chiffres de c ont été remplacés par des zéros, sinon renvoie a inchangé.
4. Écrire une fonction `main` faisant jouer les trois tours puis affichant la somme des chiffres non éliminés.

```

$ java JeuDeTrac
-----
1987654321
-----
Jet des 3 dés : 2 et 1 et 5
Quels chiffres ? 53
-----
1987604021
-----
Jet des 3 dés : 1 et 3 et 3
Quels chiffres ? 7
-----
1980604021
-----
Jet des 3 dés : 6 et 4 et 5
Quels chiffres ? 96
-----
1080004021
-----
Reste : 15

```

Ici, le 1 le plus à gauche du nombre a (chiffre de rang 10) ne peut être éliminé et n'intervient pas dans le calcul du reste, mais il permet d'obtenir un affichage homogène.