DÉPARTEMENT D'INFORMATIQUE ET DE RECHERCHE OPÉRATIONNELLE

SIGLE DU COURS : IFT 1010 (A2000)

NOM DES PROFESSEURS : Nadia El-Mabrouk, Philippe Langlais

TITRE DU COURS : Programmation I

EXAMEN INTRA

Date	: 25 octobre
Heure	: 18 :00 - 20 :00
Salle	: Z-110

DIRECTIVES PÉDAGOGIQUES :

- Toute documentation est permise.
- **Inscrivez tout de suite votre nom et code permanent dans la case** (Figure 1).
- Répondez **sur le questionnaire** en utilisant l'espace libre qui suit chaque question.

1. ....
2. ....
3. ....
4. ....
5. ....
6. ....

Total ....

FIG. 1: Inscrivez votre nom et votre code permanent ici

Question 1 (16 points) :

Cet exercice consiste à retrouver des erreurs d'exécution, de syntaxe, ou de logique.

(a) Le code suivant contient une erreur de syntaxe. Expliquer le problème (les variables sont supposées avoir été déclarées et initialisées précédemment dans le programme).

```
if (x > 0)
    sum = sum + x;
    System.out.println("x est strictement positif.");
else
    System.out.println("x n\'est pas strictement positif");
```

(b) Quelle est l'erreur engendrée par le code suivant ? Trouver une façon de corriger cette erreur, pour que le programme affiche tous les nombres entre 0 et 50.

```
int count = 50;
while (count >= 0)
{
   System.out.println(count);
   count++;
}
```

(c) Le code suivant provoque une erreur de logique. Expliquer le problème, et dire comment modifier le code pour corriger cette erreur.

```
int n, sum = 0;
for (n = 1; n < 10, n++)
     sum += n;
System.out.println("1 + 2 + ... + 9 + 10 = " + sum);
```

(d) L'instruction suivante risque de provoquer une erreur d'exécution. Expliquer le problème. Réécrire l'instruction de telle sorte à éviter cette erreur d'exécution.

```
if ( (j / i == 0) && (i > 0) )
     System.out.println("j est plus petit que i");
```

Question 2 (14 points) :

Cet exercice consiste à deviner la sortie des programmes proposés.

(a) Soient les deux instructions `if` suivantes :

```
if (vitesse > 50)
      amende = 20.00;
else if (vitesse > 70)
      amende = 40.00;
else if (vitesse > 90)
      amende = 60.00;
```

```
if (vitesse > 90)
      amende = 60.00;
else if (vitesse > 70)
      amende = 40.00
else if (vitesse > 50)
      amende = 20.00;
```

Supposons que `vitesse = 90`. Quelle est la valeur attribuée à `amende`, par l'instruction à gauche et par l'instruction à droite ? Quelle est l'instruction correcte (plus la vitesse est élevée, plus l'amende est chère) ?

(b) Soient les deux instructions `if` suivantes :

```
if (x >= 0)
      x++;
else if (x >= 1)
      x += 2;
```

```
if (x >= 0)
      x++;
if (x >=1)
      x+=2;
```

Supposons qu'au départ, $x = 2$. Quelle est la valeur finale de x, pour chacune de ces deux instructions ?

(c) Quel est l'affichage du programme suivant, pour `note='I'` ? Pour `note='B'` ? Pour `note='b'` ?

```
switch (note) {
case 'A' : points = 4; break;
case 'B' : points = 3; break;
case 'C' : points = 2; break;
case 'D' : points = 1; break;
case 'E' :
case 'I' :
case 'W' : points = 0; break;
default: System.out.println("Note invalide");
}
if (('A' <= note) && (note <= 'D'))
    System.out.println("Reussite, points gagnes = " + points);
else
    System.out.println("Echec, aucun point gagne");
```

Question 3 (15 points) :

Le programme suivant a pour objectif de comparer deux mots, et de dire si les mots sont identiques ou différents.

```
String mot1 = Keyboard.readString();
String mot2 = Keyboard.readString();
boolean test=true;

if ( mot1.length() != mot2.length() )
    test = false;

for (int i = 0; i < mot1.length(); i++)
    {
        if ( mot1.charAt(i) != mot2.charAt(i) )
             test = false; }

if (test == false)
    System.out.println ("Les deux mots sont differents");
else
    System.out.println ("Les deux mots sont identiques");
```

(a) Le code précédent n'est pas tout à fait correct. Pour certains mots différents, la sortie du programme est : "Les deux mots sont identiques". De plus, le programme peut provoquer une erreur d'exécution. Expliquer le problème. Rajouter la ou les instructions manquantes, aux endroits appropriés dans le code, pour corriger cette erreur.

(b) Tel que présenté, le programme n'est pas optimal. La boucle choisie ralentie inutilement l'exécution. Expliquer pourquoi.

(c) Réécrire un code plus efficace (exécution plus rapide), en utilisant une boucle `while`.

Question 4 (15 points) :
Un programme vous est donné instruction par instruction dans la table qui suit. Pour toutes les lignes non marquées du symbole $*$, il vous est demandé de préciser dans la colonne de droite associée :

- la valeur – après exécution de l'instruction – de la variable dont le nom est spécifié
- ou l'affichage produit par l'instruction lorsqu'il s'agit d'une instruction demandant un affichage (*ie.* contenant `System.out.println`)

<table>
<tr><td><code>int j=0, i=0, k;</code>
<code>float f=0.0f;</code>
<code>double d=0.0;</code>
<code>boolean a=true, b=true;</code></td><td>*
*
*
*</td></tr>
<tr><td><code>f = 1.5f;</code></td><td>f=</td></tr>
<tr><td><code>d = f;</code></td><td>d=</td></tr>
<tr><td><code>i = (int) d;</code></td><td>i=</td></tr>
<tr><td><code>i = 5; j = 2;</code></td><td>*</td></tr>
<tr><td><code>System.out.println(i / (float) j);</code></td><td></td></tr>
<tr><td><code>System.out.println((float) (i/j));</code></td><td></td></tr>
<tr><td><code>System.out.println(i + " + " + j + " = " + i + j);</code></td><td></td></tr>
<tr><td><code>i=4; j=3; k=1;</code></td><td>*</td></tr>
<tr><td><code>System.out.println((i>k)? (i--):(j++));</code></td><td></td></tr>
<tr><td><code>System.out.println("i=" + i + " j=" + j + " k=" + k);</code></td><td></td></tr>
<tr><td><code>i=4; j=3; k=1;</code></td><td>*</td></tr>
<tr><td><code>System.out.println((i>=4) || (k/j == 2));</code></td><td></td></tr>
<tr><td><code>System.out.println((!(i>4)) || (k/j == 2));</code></td><td></td></tr>
<tr><td><code>System.out.println((!(i > 4) && (k==1)) || ((j+k) < 3));</code></td><td></td></tr>
<tr><td><code>if ((i <= (j+k)) || (--j == 0))</code>
<code>   System.out.println("i=" + i + " j=" + j + " k=" + k);</code></td><td></td></tr>
<tr><td><code>a = true; b = true;   System.out.println((!a) || b);</code></td><td></td></tr>
<tr><td><code>a = false; b = true;   System.out.println((!a) || b);</code></td><td></td></tr>
<tr><td><code>a = true; b = false;   System.out.println((!a) || b);</code></td><td></td></tr>
</table>

Note : Ces instructions ne sont pas indépendantes, elles sont exécutées en séquence (les unes après les autres).

Question 5 (20 points) :

Écrire un programme qui lit une chaîne de caractères au clavier et qui affiche à l'écran la chaîne modifiée de la façon suivante : les consonnes sont affichées avant les voyelles ; l'ordre d'apparition de chaque lettre n'est pas modifié. Par exemple :

entrée	sortie
ananas	nnsaaa
abracadabra	brcdbraaaaa
abcdefg	bcdfgae
aaabbb	bbbaaa

Note : Il n'est pas nécessaire de créer une représentation intermédiaire pour ranger le résultat de l'application de la première règle. Vous pouvez, par exemple, faire un premier passage sur la chaîne pour traiter les consonnes, puis un second passage pour traiter les voyelles.

Pour la lecture d'un mot, utiliser la méthode `readString` de la classe `Keyboard`.

Question 6 (20 points) :
Écrire un programme qui affiche le triangle suivant :

```
              1
            1 2 1
          1 2 3 2 1
        1 2 3 4 3 2 1
      1 2 3 4 5 4 3 2 1
    1 2 3 4 5 6 5 4 3 2 1
  1 2 3 4 5 6 7 6 5 4 3 2 1
1 2 3 4 5 6 7 8 7 6 5 4 3 2 1
```

Ce programme nécessite l'utilisation de boucles `for` imbriquées.