Examen VARI1 (NFP135)

Les documents imprimés sur papier sont autorisés, smart-phone/tablettes/ordinateurs interdits.

Note : chaque exercice vaut 2.5 points, pour un total de $8 \times 2.5 = 20$.

Exercice 1 Écrire un programme (`java` ou `processing`) dans lequel on initialise un tableau de 7 entiers (vous pouvez écrire ces 7 valeurs en dur dans le code). Modifier le programme pour le faire afficher la plus grande valeur du tableau. **Attention** : le calcul de la valeur maximale doit fonctionner pour tout tableau de 7 cases, pas uniquement pour les valeurs que vous avez choisies.

Exercice 2 Écrire une fonction Java `imcCorrect(double kg, double m)` qui prend deux arguments de type `double` : le poids en kg et la taille en mètres. Cette fonction doit d'abord calculer la valeur de l'IMC (Indice de Masse Corporelle). La valeur de l'IMC est le poids divisé par la taille au carré ; on utilise la formule IMC= $\frac{\texttt{kg}}{\texttt{m}^2}$. Finalement, cette fonction doit renvoyer un `boolean` qui indique si l'IMC calculé peut être considéré comme correct ou pas. Un IMC correct est un IMC supérieur à 20 et inférieur à 25.

Exercice 3 Écrire une fonction Java `puissance(...)` qui prend deux arguments x et n de type `int` et qui renvoie la valeur x^n. Le résultat sera un `int`. L'en-tête (ou la signature) de la fonction est :

```
int puissance(int x, int n)
```

Attention : si n est inférieur à zéro, il faut renvoyer -1 et afficher « impossible de calculer cette puissance avec un exposant négatif ».

Exercice 4 Écrire un programme `Exo4.java` qui calcule le nombre de valeurs positives (ou nulles) et le nombre de valeurs négatives dans un tableau d'entiers `t` déclaré et initialisé au début du programme. Le résultat doit être stocké dans deux variables globales (attention statiques !) `nbPosit` et `nbNegat`.

Exercice 5 Écrire un programme complet `Java` qui permet de lire 5 entiers naturels à partir d'un fichier texte `in.txt` (utiliser soit la classe `BufferedReader` soit la classe `Scanner`). On suppose que ce fichier comporte un entier par ligne, donc un total de 5 lignes. Afficher la somme des entiers positifs trouvés dans le fichier

Exemple : pour le fichier à droite, le programme devrait afficher `15`.

```
in.txt
------
3
-4
2
10
-40
```

Exercice 6 Écrire une fonction d'en-tête

```
int lePlusGrandDiviseur(int n)
```

qui renvoie le plus grand diviseur de n. Par exemple, il faut renvoyer 50 pour $n = 100$, 3 pour $n = 9$, 11 pour $n = 55$, ou 7 pour $n = 7$.
Note *: la condition* `if (x%y==0)` *passe uniquement si y est un diviseur de x.*

Exercice 7 Écrire une fonction d'en-tête

```
int nbPremiers(int[] tab)
```

qui renvoie le nombre de nombres premiers dans un tableau `tab`. Un nombre n est premier s'il a que deux diviseurs : 1 et n. Il faut se servir de la fonction de l'exercice précédent.

Exercice 8 Question de théorie :

A.) Que représente la figure à droite ? Décrire en quelques lignes (maximum 10) les interactions et les modules dans cette figure. Par exemple, expliquer ce que représente les flèches des programmes utilisateurs vers l'interface utilisateur, ou détailler l'expression « mémoire RAM (vive ou virtuelle) » **[0.7pt]**

B.) Donner six commandes `Linux` (resp.) pour :

- afficher le contenu d'un fichier `Toto.java`.
- compiler le programme `Toto.java` et lancer ce programme
- éffacer le fichier `Toto.java`
- se placer dans le dossier parent (c.-à.-d, père)
- afficher les fichiers du dossier courant
- afficher la manuel de la commande `cat`.

[1.8pt]