Examen VARI1 (NFP135)

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

Cet examen comporte 8 exercices mais je vous propose de **résoudre uniquement 7 exercices** à votre choix parmi les 8. Je vais donner 3 points pour chaque exercice choisi, pour un total de maximum 21 (dont 1 point bonus).

Exercice 1 Écrire une fonction Java `division(...)` qui prend deux paramètres de type `double` et qui renvoie le résultat de la division de ces deux paramètres. Naturellement, le résultat sera un `double`. On dit que l'en-tête (ou la signature) de la fonction est :

double division(**double** x, **double** y)

Attention : si $y = 0$, il faut renvoyer 10.000.000 et afficher « impossible de faire une division par zéro ».

Exercice 2 Écrire une fonction d'en-tête

int nbNotesValides(**int**[] tab)

qui renvoie le nombre de notes valides dans un tableau `tab`. On dit qu'une note est valide si elle est supérieure ou égale à zéro et inférieure ou égale à 20.

Exercice 3 Écrire un programme `Exo3.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 4 Écrire un programme complet `Java` qui permet de lire 5 entiers naturels à partir d'un fichier texte `in.txt` (utiliser la classe `BufferedReader`). On suppose que ce fichier comporte un entier par ligne, donc un total de 5 lignes. Afficher le produit des entiers non-nuls trouvés dans le fichier

Exemple : pour le fichier à droite, le programme devrait afficher `240`.

in.txt
```
3
4
2
10
0
```

Exercice 5 Écrire une fonction d'en-tête

int noteLaPlusFréquente(**int**[] notes)

qui renvoie la note/valeur qui apparait le plus souvent dans un tableau `notes` de notes ; bien évidement, les notes sont toutes entières et comprises entre 0 et 20. Par exemple, pour `notes={11, 15, 3, 3, 5, 8, 15, 9, 15}`, la fonction devrait renvoyer 15, car 15 est la valeur/note la plus fréquente.
Indication : Vous pourriez penser à construire un nouveau tableau qui compte le nombre d'apparitions de chaque note entre 0 et 20 ; à la fin, il faut juste renvoyer la case de la valeur maximale de ce deuxième tableau.

Exercice 6 Écrire une méthode (fonction qui ne renvoie rien) `espacesÉtoiles(int nbEspaces, int nbÉtoiles)` qui affiche `nbEspaces` espaces suivis de `nbÉtoiles` étoiles.
Dans une deuxième étape, écrire une méthode `losange(int taille)` qui affiche un losagne sous le format exemplifié à droite pour `taille=7`. On suppose que la taille est toujours une valeur impaire qui indique le nombre de lignes. Le nombre d'espaces affichés à la première ligne est $\frac{\texttt{taille}-1}{2}$. À la deuxième ligne, on doit afficher $\frac{\texttt{taille}-1}{2} - 1$ espaces ; à la 3ème ligne, on affiche $\frac{\texttt{taille}-1}{2} - 2$, espaces, etc. Le nombre d'espaces diminue jusqu'au moment où on arrive au milieu du losange (la ligne numéro $\frac{\texttt{taille}-1}{2} + 1$), pour remonter à partir de cette ligne au milieu. Il faut appeler `espacesÉtoiles(..., ...)` plusieurs fois (plus exactement `taille` fois) dans la fonction `losange(...)`.

```
   *
  ***
 *****
*******
 *****
  ***
   *
```

Exercice 7 Cet exercice porte sur une méthode très simplifiée pour compresser les données. L'idée de base est la suivante : si on a une chaine de caractères dans laquelle une lettre x apparait n fois, il faut remplacer les n apparitions de x avec nx. Exemple : au lieux d'écrire $aaaaaaa$, il vaut mieux écrire $7a$, iii devient $3i$, ou $zzzttttb$ devient $3z4tb$. Par contre, $ssii$ reste $ssii$, car on gagne aucun caractère si on remplace $ssii$ avec $2s2i$. Écrire un programme qui demande à l'utilisateur de saisir une chaine de caractères et qui affiche la version compressée de cette chaine de caractères. On suppose que l'utilisateur saisit uniquement des lettres, sans aucun chiffre.

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) » **[1.2pt]**

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

- se placer dans le dossier `/home/licencep/`
- se placer dans le dossier parent (c.-à.-d, père)
- afficher les fichiers du dossier courant
- afficher le contenu d'un fichier `Scoop.java`.
- afficher la manuel de la commande `ls`
- afficher toutes les lignes du fichier `Scoop.java` qui contiennent le mot «`toto`».

[1.8pt]