
Examen VARI1 (NFP135)

Les documents imprimés sur papier sont autorisés ; toute communication électronique (sauf pour teams) est interdite.

Déposer à la fin de la séance un fichier texte qui représente votre copie au site indiqué ci-après : <http://cedric.cnam.fr/~porumbed/vari1/exam/>

Chaque exercice vaut 3 points sauf le dernier qui vaut 2 points.

Exercice 1 Écrire un programme (java ou processing) dans lequel on initialise un tableau de 7 entiers (vous pouvez mettre en dur les valeurs que vous souhaitez dans ce tableau). Modifier le programme pour le faire afficher la plus petite valeur dans ce tableau. **Attention** : le calcul de la valeur minimale doit fonctionner pour tout tableau de 7 cases, pas uniquement pour les valeurs que vous avez choisies.

Exercice 2 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 3 Écrire une fonction d'en-tête

```
int nbPersonnesImpotsZero(int[] revenus)
```

qui renvoie le nombre de personnes qui payent un impôt de zéro étant donné les revenus stockés dans le tableau `revenus`. On considère un pays imaginaire où 1) une personne qui gagne plus de 20000 euros doit payer des impôts, et 2) une personne qui gagne moins de 10000 euros doit payer une forme d'impôt négatif, ex, on lui offre une prime pour l'emploi. Bref, on parle d'un impôt de zéro uniquement pour un revenu entre 10000 et 20000 euros compris.

Exercice 4 Écrire un programme `Exo4.java` qui calcule le nombre de valeurs positives, nulles et négatives dans un tableau de valeurs réelles `t` déclaré et initialisé au début du programme. Le résultat doit être stocké dans trois variables globales (attention statiques!) `nbPosit`, `nbZeros` et `nbNegat`. À la fin, afficher « Attention, il n'y a que des valeurs de même type » si une des conditions suivantes est vérifiée : (1) la tableau contient que des zéros, (2) la tableau contient que des valeurs positives, (3) la tableau contient que des valeurs négatives.

Exercice 5 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 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 lePlusGrandDiviseurPremier(int n)
```

qui renvoie le plus grand diviseur premier de n . J'insiste qu'il faut renvoyer un diviseur premier, voir aussi l'exercice 7 du TP12.

Exemples, il faut renvoyer 3 pour $n = 12$, 5 pour $n = 100$, 3 pour $n = 9$ ou 11 pour $n = 55$.

Note : la condition `if (x%y==0)` passe uniquement si y est un diviseur de x .

Exercice 7 Question de théorie :

A.) Décrire en moins de 50 mots ce que la figure à droite représente.

[1pt]

B.) Donner quatre commandes Linux (resp.) pour :

- afficher le contenu d'un fichier `Toto.java`.
- compiler le programme `Toto.java`
- se placer dans le dossier parent (c.-à.-d, père)
- afficher les fichiers du dossier courant

[1pt]

