
Examen VARI1 (NFP135)

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

Chaque exercice vaut 2.5 points.

Exercice 1 Écrire un programme `processing` pour tracer :

- une ligne noire qui relie les points (50, 50) et (70, 70) ;
- un cercle rempli de couleur rouge ;
- un rectangle non-rempli qui couvre tous les dessins précédents.

Exercice 2 Écrire une fonction `vitesseMoyenne(...)` sous `Java` or `Processing` (votre choix, car c'est presque la même chose). Cette fonction prend deux arguments de type `double` : une distance (en km) et une durée (en heures). Cette fonction doit renvoyer un `double` qui représente la vitesse moyenne, c.à.d., la distance divisée par la durée. On dit que l'en-tête (ou la signature) de la fonction est :

```
double vitesseMoyenne(double distance, double durée)
```

Le corps de cette fonction (votre réponse) devrait avoir une seule ligne pour faire le calcul $\frac{\text{distance}}{\text{durée}}$ et renvoyer le résultat avec `return`.

Exercice 3 Écrire une fonction Java `noteValide(...)` qui prend un seul argument de type `double` ; cette fonction doit renvoyer un `boolean` qui indique si l'argument représente une note valide ou pas. On dit que l'en-tête (ou la signature) de la fonction est :

```
boolean noteValide(double x)
```

Note : Une note est valide si elle est supérieure ou égale à zéro et inférieure ou égale à 20.

Exercice 4 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 5 Écrire une fonction d'en-tête

```
int nbNotesValides(int[] tab)
```

qui renvoie le nombre de notes valides dans un tableau `tab`. Vous pouvez appeler la fonction `noteValide(...)` écrite à l'exercice 2.

Exercice 6 Écrire un programme `Exo6.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 7 Écrire un programme qui demande à l'utilisateur de saisir 7 valeurs. Calculer et afficher le produit maximal entre deux des valeurs saisies. Par exemple :

- Il faut afficher 100 si l'utilisateur saisit 4, 10, 3, 10, 6, 8, 9
- Il faut afficher 110 si l'utilisateur saisit 4, 11, 3, 10, 6, 8, 9
- Il faut afficher 800 si l'utilisateur saisit 100, 1, 3, 1, 6, 8, 7

Exercice 8 Question de théorie :

A.) Que représente la figure à droite? Décrire en moins de 100 mots 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) »

[1pt]

B.) Donner une commande Linux (resp.) pour :

- afficher les fichiers du dossier courant
- afficher le contenu d'un fichier `Toto.java`.
- compiler le programme `Toto.java` et lancer ce programme
- se placer dans le dossier `/toto/licencep/`
- afficher la manuel de la commande `cat`.

[1pt]

C.) Décrire en moins de 25 mots la différence entre une variable locale et une variable globale!

[0.5pt]

